Switch 4007 Implementation Guide

Release 3.0.5

http://www.3com.com/

Part No. 10013673
Published May 2000

3Com Corporation
5400 Bayfront Plaza
Santa Clara, California
95052-8145

CONTENTS

5 MANAGING THE CHASSIS POWER AND TEMPERATURE

7 PHYSICAL PORT NUMBERING

8 ETHERNET

9 BRIDGE-WIDE AND BRIDGE PORT PARAMETERS

10 CLASS OF SERVICE (COS)

11 IP MULTICAST FILTERING WITH IGMP

12 TRUNKING

13 RESILIENT LINKS

14 VIRTUAL LANS (VLANS)

15 PACKET FILTERING

17 VIRTUAL ROUTER REDUNDANCY PROTOCOL (VRRP)

20 IPX ROUTING

21 APPLETALK ROUTING

22 QoS and RSVP

23 DEVICE MONITORING

A TECHNICAL SUPPORT

INDEX

ABOUT THIS GUIDE

This *Switch 4007 Implementation Guide* provides information that you need to understand and use features of the Switch 4007 after you install it and attach it to your network.

Before you use this guide:

- Install your switch chassis and modular components. See the *Switch 4007 Getting Started Guide* and the individual module *Quick Start* guides for installation procedures, cabling information, and environmental information.
- Read Chapter x of this guide, to learn more about the Switch 4007 Management Module.
- Read Chapter 1 of this guide, which provides an overview of the configuration process.
- Become familiar with the *Switch 4007 Command Reference Guide* which documents the commands that you use to configure and manage Layer 2 Switching Modules and Multilayer Switching Modules through a built-in, menu-driven interface called the Administration Console.

Audience

This guide is intended for the network administrator who is responsible for configuring, using, and managing the Switch 4007. It assumes a working knowledge of local area network (LAN) operations and familiarity with communications protocols that are used on interconnected LANs.

Scope of this Guide

The information in this guide pertains to Release 3.0.5 software.

Switch 4007 modules are pre-loaded with software at the factory. However, the software that was loaded on the components that you received may be an earlier release. Connect to each module and use the `module display` *command to determine what release is loaded. Go to the 3Com Web site (*`http://support.3com.com`*) to download the latest software.*

If the information in the release notes that correspond with the software on your modules differs from the information in this guide, follow the instructions in the release notes.

The Switch 4007 software and management interfaces are built from CoreBuilder® 9000 switch technology. In Switch 4007 software releases 3.0.0 and 3.0.5, the prompts and displays in all interfaces may indicate this heritage.

Conventions

Table 1 and Table 2 list icon and text conventions that are used throughout this guide.

Table 1 Icons

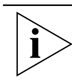
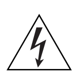
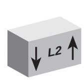
Icon	Type	Description
	Information note	Information that describes important features or instructions
	Caution	Information that alerts you to potential loss of data or potential damage to an applications, system, or device
	Warning	Information that alerts you to potential personal injury
	Layer 2 switch	In figures, a switch or module that can perform Layer 2 functions.
	Layer 3 switch	In figures, a switch or module that can perform both Layer 2 and Layer 3 functions.

Table 2 Text Conventions

Convention	Description
`Screen displays`	This typeface represents information as it appears on the screen.
Syntax	The word "syntax" means that you evaluate the syntax provided and then supply the appropriate values. Example: To set the system date and time, use the following syntax: `mm/dd/yy hh:mm:ss: xM`
`Commands`	The word "command" means that you type the command exactly as shown in the text and then press Return or Enter. Commands appear in bold. Example: To remove an IP interface, enter the following command: **`ip interface remove`** *This guide always gives the full form of a command in uppercase and lowercase letters. However, you can abbreviate commands by entering only enough characters to differentiate each command. Commands are not case sensitive.*
The words "enter" and "type"	When you see the word "enter" in this guide, you must type something, and then press Return or Enter. Do not press Return or Enter when an instruction simply says "type."
Keyboard key names	If you must press two or more keys simultaneously, the key names are linked with a plus sign (+). Example: Press Ctrl+Alt+Del.
Words in *italics*	Italics are used to: ■ Emphasize a point ■ Denote a new term when it is defined in text

Switch 4007 Documentation

The Switch 4007 documentation set is comprised of many different titles.

Some Switch 4007 documents use the product name "CoreBuilder 9000" in their titles due to the heritage of the product line.

Documents are available in three formats:

- **Paper Documents** — A few hardcopy documents are shipped with your chassis. These mainly pertain to chassis installation.
- **Documents on CD-ROM** — A *Switch 4007 Software and Online Manuals CD* is shipped with your chassis. It contains online (PDF) versions of *all* Switch 4007 documents (software and hardware guides), except for release notes (which you must download from the 3Com Web site).
- **World Wide Web** — All user guides and release notes are available in Adobe Acrobat Reader PDF or HTML format from the 3Com Web site at: `http://support.3com.com/`

Although they do not ship with your chassis in paper form, you can order printed and bound copies of the Switch 4007 Implementation Guide *and* Switch 4007 Command Reference Guide by using *3Com order number 3C16803.*

*Switch 4007 Release Notes are **not** shipped in paper form and are **not** included on the Switch 4007 Software and Online Manuals CD. You must download all release notes from the 3Com Web site.*

Documentation Comments

Your suggestions are very important to us. They help us to make our documentation more useful to you.

Please send e-mail comments about this guide to:

sdtechpubs_comments@ne.3com.com

Please include the following information when you comment:

- Document title
- Document part number (found on the front or back page of each document)
- Page number (if appropriate)

Example:

Switch 4007 Implementation Guide

Part Number 10013673

Page 25

Year 2000 Compliance

For information on Year 2000 compliance and 3Com products, visit the 3Com Year 2000 Web page:

http://www.3com.com/products/yr2000.html

I

UNDERSTANDING YOUR SWITCH 4007 SYSTEM

1 CONFIGURATION OVERVIEW

This chapter lists the basic requirements for building a functional Switch 4007, summarizes the system architecture, describes the management interface options, and provides a general configuration procedure.

The chapter covers these topics:

- Physical Configuration Requirements and Options
- System Architecture
- Management Options
- Management Access
- System Configuration Process

The Switch 4007 chassis supports frame-based technology with the Gigabit Ethernet switch fabric module and Fast Ethernet and Gigabit Ethernet interface modules (Layer 2 Switching Modules and Multilayer Switching Modules).

For overview information about all Switch 4007 components, see the Switch 4007 Getting Started Guide*.*

Physical Configuration Requirements and Options

The Switch 4007 chassis provides the following features:

- Bays for two power supplies to provide from 820 watts to 1860 watts, depending on the type and quantity of installed modules.
- Power fault-tolerant mode so that you can reserve the power of a single power supply to act as backup if one of the power supplies fails.
- Four exhaust fans (in one fan tray) to make sure that the chassis maintains the optimal temperature for operation.
- Two management module slots that do not take up space for switch fabric modules or interface modules (there are seven other slots for these types of modules).
- One switch fabric module slot (leaving six slots for LAN interface modules).

Requirements

Building a functional Switch 4007 requires the following items:

- One Switch 4007 chassis
- A sufficient number of power supplies to support all installed components
- A sufficient number of cooling fans to support all installed components
- One Management Module

The Management Module may be referred to as the Enterprise Management Engine (EME) in this guide or in the product's management interfaces. This is because the heritage of the Switch 4007 product line is the CoreBuilder® 9000 product line.

- One Gigabit Ethernet (GEN) Switch Fabric Module (select a 24-port module or 9-port module)
- One or more interface modules that connect to your LAN. Examples:
  - 36-port 10/100BASE-TX Fast Ethernet Layer 2 Switching Module
  - 9-port 1000BASE-SX Gigabit Ethernet Layer 2 Switching Module
  - 4-port GBIC Gigabit Ethernet Multilayer Switching Module

Multilayer Switching Modules may be referred to as Layer 3 Switching Modules either in this guide or in the product's management interfaces. This is because the heritage of the Switch 4007 product line is the CoreBuilder® 9000 product line.

Options The Switch 4007 offers the following options for high device availability:

- You can install a second Management Module for redundant management and chassis controller functions
- The switch provides intelligent N+1 power and environmental management systems. See the *Switch 4007 Getting Started Guide*.

Order of Installation Activities 3Com recommends that you install and configure Switch 4007 items in the following order:

1. Chassis components such as power supplies and fan trays (unless they were already installed at the factory)
2. Management modules (unless they were already installed at the factory)
3. Switch fabric module (unless it was already installed at the factory)
4. LAN switching modules (unless they were already installed at the factory)

All Switch 4007 modules support hot-swap functionality so that the system can respond to dynamic changes.

For installation procedures, as well as restrictions and recommendations for module placement in the chassis, see the Switch 4007 Getting Started Guide *or the appropriate module* Quick Start Guide*. For module software compatibility information, see the* Switch 4007 Release Notes*.*

System Architecture

The Switch 4007 system uses separate channels for network traffic and management traffic:

- A separate 10 Mbps management LAN (MLAN) carries management traffic to and from the EME, which acts as the single point of contact for all management traffic in the chassis. For example, an SNMP request to a certain module is first received by the EME and then communicated by proxy from the EME to the module.
- A series of channels on the chassis backplane carry network traffic. These channels radiate from the switch fabric module slot to every other switching module slot in the chassis. The ports on the rear of an installed module connect it with the backplane. LAN interface modules pass data through the backplane to the switch fabric module. The switch fabric module may then direct the data to the backplane to other modules in the chassis (which may then direct it out their front panel ports) or it may direct it through one of its front panel ports (to another device).

Management Options

Depending on the task you want to accomplish or what kind of information you need, you have the following management options on the Switch 4007:

- Management Module Console
- Switching Module Administration Console
- Web Management software (allows you to configure both Management Module and Switching Module options)
- SNMP Management software

Each option is described in the following sections.

Management Module Console

The command line interface (CLI) of the Management Module serves as the single point of contact for managing the entire system. It manages all system-level functions such as login management, IP and SNMP connectivity, software downloads to all modules in the chassis, system inventory management, and power management.

You can connect to the EME in the following ways:

- RS-232 Terminal serial port
- RS-232 Modem (auxiliary) serial port
- RJ-45 10BaseT Ethernet port

With the serial ports you can manage your system locally through a terminal connection or remotely using a Telnet or modem connection.

If you want to use SNMP-base management software, you must connect a cable from the module's Ethernet port to your network infrastructure.

You must log in to the Management Module Console prior to connecting to the Administration Console of a switching module.

For more information about the Management Module and its CLI, see Part I of this guide.

Switching Module Administration Console

Each switch fabric module and LAN interface module has a built-in management interface called the Administration Console. It is a menu-driven CLI that provides module-specific menus and parameters. (i.e., the menus are different between Layer 2 Switching Modules and Multilayer Switching Modules). You use this interface to configure all parameters for a particular module and to display module statistics and summary configurations.

To access the Administration Console on a particular switching module in the chassis, you must first log in to the Management Module and then connect to the module slot. You can view the Administration Console from a terminal, a PC, a Macintosh, or a UNIX workstation.

For more information about the features and options in the Administration Console, see the chapters in Part II of this guide. For information on specific commands, see the *Switch 4007 Command Reference Guide*.

Web Management software

A suite of HTML-based applications are shipped with your Switch 4007 chassis package. The suite consists of embedded Web Management software applications as well as other tools that you can install:

- **Embedded Web Management applications** — Use the embedded Web Management applications for most of your device configuration and management tasks. You can manage a single port or device, or, using multiple windows, you can manage multiple devices. This software, which is part of the system software image, contains:
  - **WebConsole** — An HTML-based set of configuration forms.
  - **DeviceView** — A Java-based application that displays a real-time image of the device. You can manage each port, module, or system by clicking the part of the image that you want to manage.
  - **Performance features** — Dynamic monitoring through graphing of QoS statistics and Ethernet interfaces.
  - **Help** — Access to the configuration form on which you set up the installable Help files as well as access to links to support information on the 3Com Web site.
- **Installable tools** — Install these optional tools on your workstation from the 3Com Web site:
  - **DeviceView accessories** — To set up e-mail notification for Status Logging

- **WebManage Framework** — To group your access links to the devices that you manage
- **Filter Builder** — To create and test filters for packets on your switch
- **Form-specific Help** — To get more information about WebConsole, DeviceView, and Performance forms

After you have set up your IP address for the Switch 4007 system, you can access Web Management applications directly in your Web browser by entering the system IP address.

For information about setting up an IP address, see the chapters in Part I in this guide.

For more information about Web Management applications, browser requirements, and other aspects, see the *Switch 4007 Getting Started Guide.*

SNMP-Based Network Management Overview

For a more comprehensive approach to network management, you can use an external application that uses the Simple Network Management Protocol (SNMP) to communicate with the Switch 4007. As part of the IP protocol suite, SNMP is the standard management protocol for multivendor networks. SNMP supports transaction-based queries so that the protocol can format messages and transmit information between reporting devices and data-collection programs.

In order for SNMP requests to reach the Switch 4007, you must connect a cable from the Ethernet port on the Management Module to your network infrastructure (for example, a hub or a switch). You must also assign an IP address to the Ethernet port.

The Management Module acts as an agent in an SNMP-managed environment. The agent polls modules in the chassis for information, responds to SNMP requests, and generates SNMP traps.

You must configure SNMP settings in two or more places: the Management Modules and individual switching modules.

Management Access

After you assign a unique IP address to the Management Module, you can access the system through the IP interface in one of the following ways:

- Through up to four remote Telnet sessions to the Administration Console. (You can establish up to four remote (Telnet) sessions and one terminal console session simultaneously.)
- Through an SNMP-based network management application.

You can access the Switch 4007 system locally through a terminal connection or remotely using a modem or an IP connection. Table 3 describes these different methods:

Table 3 Connecting to the Management Module

Access Method	Allows you to	EME Connection
Terminal	■ Connect directly to the Management Module CLI	RS-232 serial port
Modem	■ Access the Management Module CLI by dialing in from remote sites	RS-232 auxiliary serial port
Internet Protocol (IP)	■ Access the Management Module CLI using Telnet commands ■ Use an external SNMP management application to communicate with the Switch 4007 SNMP agent.	10BASE-T Ethernet port that is assigned with an IP interface

Terminal Port Access

Direct access to the management interfaces through the terminal serial port is often preferred because you can remain on the system and monitor it during system reboots. In addition, certain error messages are sent to the serial port, regardless of the interface through which the associated action was initiated.

A Macintosh or PC attachment can use any terminal emulation program for connecting to the terminal serial port. A workstation attachment under UNIX can use an emulator such as TIP.

For more information about terminal port configuration options, see the chapters in Part I of this guide.

Modem Port Access

You can access the management interfaces from your PC or Macintosh using an external modem attached to the modem serial port. The system transmits characters that you have entered as output on the modem port. The system echoes characters that it receives as input on the modem port to the current Administration Console session. The console appears to be directly connected to the external modem.

For more information about modem configuration options, see the chapters in Part I of this guide.

Access Levels

The Management Module CLI and the Administration Console CLI support three access levels so that you can provide different levels of access for a range of users, as described in Table 4.

You configure access levels and passwords on the Management Module because that is the first point of entry. These conditions apply to both the Management Module CLI and the Administration Console CLI of switching modules. For example, if a user logs in to the Management Module with `Write` privileges, then the user connects to any module's Administration Console at the same level.

Table 4 Access Levels

Access Level	For Users Who Need to	Allows Users to
Administer	Perform system setup and management tasks (usually a single network administrator)	Perform system-level administration (such as setting passwords, loading new software, and so on)
Write	Perform active network management	Configure network parameters (such as setting bridge aging time)
Read	Only view system parameters	Access only *display* menu items (display, summary, detail)

Only one user at a time can log in with Administer privileges.

For additional information about user login functions, see the chapters in Part I of this guide.

System Configuration Process

This guide assumes that you have completed the physical installation process for all items in the Switch 4007 chassis successfully and that you are ready to begin configuring and managing your system. Use this *Switch 4007 Implementation Guide* together with the *Switch 4007 Command Reference Guide* to gain helpful conceptual and practical information.

A version of operational software for each module is installed at the factory. However, this version may not be the latest version available.

You can download the latest software from the 3Com Web site (`http://support.3com.com/`*). When you download software, be sure to also download the corresponding release notes to learn about software compatibility requirements among modules in the chassis, known problems with the software, and other issues.*

Because the software in a module boots automatically as long as power is available, a module is immediately ready to configure and manage according to your network needs. See “Configuration Procedure” next in this chapter for a list of required and recommended steps.

Configuration Procedure

Follow the steps that apply to your system configuration and network needs and ignore the steps that do not apply.

Configure the Management Module

You must configure the Management Module with certain parameters before you access the Administration Console of any switch fabric module or interface module and before you access the system through an external Simple Network Management Protocol (SNMP) application. See Chapter X in this guide for more information.

Configure Each Switching Module

1. Configure basic management or physical link parameters. One or more of the following topics may apply to each module:
   - **Module parameters** — To manage nonvolatile data (nvData), display your module status information, display warning messages for certain module conditions, and reset baseline counters, see Chapter 6.
   - **Physical port numbering** — To learn the port numbering rules and understand the effects of adding or removing modules, see Chapter 7.

- **Ethernet** — To label Ethernet ports, set the port mode, enable flow control, and control autonegotiation and other settings, see Chapter 8.
- **Bridge-wide and bridge port parameters** — To set parameters for Spanning Tree Protocol, IPX SNAP translation, and IP fragmentation, and address aging options, see Chapter 9.
- **Trunks** — To increase the bandwidth and resiliency between two points, you can aggregate many individual links into a single logical link called a *trunk*. Configure trunks before you define VLANs. For more information, see Chapter 12.

2 Define virtual LANs (VLANs).

To create logical workgroups in Layer 2 Switching Modules, you can define port-based VLANs. Be sure to define trunks before you define VLANs.

To create logical workgroups in Multilayer Switching Modules, you can define port-based, protocol-based, or network-based VLANs, and set related modes. On Multilayer Switching Modules, you must define VLANs before you define routing interfaces.

For more information about VLANs, see Chapter 14.

3 Configure routing interfaces and set related parameters.

You can use the following protocols to configure routing interfaces and set related parameters:

- **IP** — See Chapter 16.
- **IP Multicast** — See Chapter 18
- **Open Shortest Path First (OSPF)** — See Chapter 19
- **IPX** — See Chapter 20.
- **AppleTalk** — See Chapter 21.

4 Configure more advanced traffic control features: packet filters, Quality of Service (QoS), and the Resource Reservation Protocol (RSVP).

To improve LAN performance, shape traffic flows, or implement security controls with standard, custom, predefined, and port group packet filters, see Chapter 15.

To classify, control, and prioritize traffic where available bandwidth is low and your network is carrying time-sensitive or business-critical information, use the QoS and RSVP features. For more information, see Chapter 22.

5 Repeat steps 1through 4 (as applicable) for each module in your system.

6 Take advantage of device monitoring features as you monitor network operations.

You can use device monitoring features such as event logging, baselining, and roving analysis to analyze your network periodically and identify potential network problems before they become serious problems. To test and validate paths in your network, use tools like ping and traceRoute. SNMP and Management Information Bases (MIBs) provide ways to collect performance data on your network. For more information about these features, see Chapter 23.

II

UNDERSTANDING THE MANAGEMENT MODULE

2 OVERVIEW OF THE MANAGEMENT MODULE

This chapter introduces the Management Module for the Switch 4007. It addresses these topics:

- Module Overview
- Module Components
- Module Functions
- Impact on the Network

Before You Start

Before you install the Management Module or begin to explore the options on the Management Module, make sure that you have properly unpacked and installed your chassis in a rack, on a shelf, or on a table and that you have read the following documents:

- *Switch 4007 Getting Started Guide*
- *Enterprise Management Engine Quick Start Guide for the CoreBuilder® 9000 Enterprise Switch*
- *Enterprise Management Controller Quick Start Guide for the CoreBuilder 9000 Enterprise Switch*
- Release Notes for appropriate modules or groups of modules at specific software releases.

The heritage of the Switch 4007 product line is the CoreBuilder 9000 product line. Some documents have not been rebranded and continue to use "CoreBuilder 9000" and "Enterprise Management Engine" in their titles.

In addition, be sure to review the first two chapters in this guide for an overview of system functions.

Module Overview

The Management Module is an SNMP-based network management module that:

- Provides chassis controller functions (such as power and temperature monitoring), as well as management functions (such as collecting and sending SNMP traps).
- Provides a central point of contact for management of all components in the chassis.
- Manages power use in the chassis by:
  - Preventing newly installed modules from receiving power when there is not enough power available
  - Allowing you to prioritize the order in which modules power off (if there is insufficient power available)
  - Allowing you to implement fault-tolerant power, which allows the chassis to reserve some of its power capacity to protect against a power supply failure
- Exchanges information with all other modules through the 10 Mbps management LAN, which keeps management traffic separate from network traffic.

Module Components

The Management Module consists of the following two components:

- System Management Component (SMC)
- System Controller Component (SCC)

Both components share the same in-rush current, clocks, and backplane interfaces. The remaining circuits in both components are separated and are controlled by two dedicated CPUs.

The four-character LED display on the Management Module front panel shows the status of the SMC component. The Active and Stndby (Standby) LEDs indicate the SCC status on the front panel also.

Module Functions

The Management Module provides the following management and control capabilities:

- **Configurations** — When you are logged in with `Administer` access, you can configure the Management Module and monitor the chassis environment.
- **Management Module standby support** — Redundant Management Modules share configuration information, so that a standby Management Module has the same configuration as the active Management Module. This capability enables a standby Management Module to become functional if the active Management Module fails.
- **Inventory** — The Management Module provides an inventory of chassis contents, including fans and power supplies. The inventory lists current software revisions for all installed modules. The inventory system also supports a scratchpad feature so that you can add custom information to the Management Module display.
- **Power management** — You can manage how the chassis reacts to low power situations. The chassis can also provide fault-tolerant power, which protects the system against power supply failures.
- **File System** — A storage area on the Management Module stores the event log and software configuration files. The file system also acts as a temporary storage area for software images that are being downloaded to it or any other module in the chassis.
- **In-band and out-of-band software download** — The Management Module provides both in-band and out-of-band software download capability. An in-band download uses TFTP (Trivial File Transfer Protocol) through a network connection. An out-of-band download uses XMODEM software and the RS-232 serial port on the front panel of the Management Module. The Management Module allows you to download software to multiple modules using a single command.
- **SNMP support** — Simple Network Management Protocol (SNMP) is a standard protocol for managing multivendor networks. The Management Module acts as an agent in an SNMP-managed environment. The agent responds to SNMP requests and generates SNMP traps.

- **Telnet support** — You can connect a Management Module to any other Telnet device. The Management Module also supports incoming Telnet sessions so that you can manage an Management Module or another module from a workstation with Telnet support or from another Management Module.
- **Web Management support** — You can monitor and manage the Management Module through the CoreBuilder 9000 Web Management suite of applications.

Impact on the Network

The Management Module generates packets on the network when it:

- Establishes and maintains a Telnet session, either as a client or a server.
- Translates an IP address to a MAC address using the Address Resolution Protocol (ARP).
- Initiates or responds to a `ping` command.
- Responds to a Simple Network Management Protocol (SNMP) request.
- Sends SNMP traps.
- Performs an in-band download using the Trivial File Transfer Protocol (TFTP).
- Sends REM and CRS information.
- Generates MAC frames.

3 INSTALLING MANAGEMENT MODULES

This chapter describes installation and setup procedures for the Switch 4007 Management Modules. The sections are:

- Installing Modules
- Creating a Redundant Configuration
- Verifying Management Module Operation
- Making Management Connections
- EME Technical Specifications

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Before You Start

Before you install a Management Module (EME), make sure that you have properly unpacked and installed your chassis in a rack, on a shelf, or on a table and that you have read the following documents:

- *Switch 4007 Getting Started Guide*
- *Enterprise Management Engine Quick Start Guide for the CoreBuilder® 9000 Enterprise Switch*
- Release Notes for appropriate modules or groups of modules at specific software releases.

The heritage of the Switch 4007 product line is the CoreBuilder® 9000 product line. Some documents have not been rebranded and continue to use "CoreBuilder 9000" and "Enterprise Management Engine" in their titles. All relevant documents are available from the Switch 4007 Software and Online Manuals CD or from the 3Com Web site.

In addition, be sure to review the earlier front matter and chapters in this guide for an overview of system functions and management module functions.

Installing Modules

One Management Module (EME) is required in each chassis. A second is optional for redundancy. The slots for management modules are obviously in the upper portion of the chassis due to their smaller size. 3Com recommends that you install all Management Modules prior to adding switching modules.

This section may not apply to you if you ordered a starter kit that has the Management Module pre-installed at the factory and if you do not intend to install other management modules. However, read this section if you are assembling separately ordered chassis components or if want to install a second Management Module.

For a detailed module installation procedure, see the *Enterprise Management Engine Quick Start Guide for the CoreBuilder 9000 Enterprise Switch.* This document is available from the Switch 4007 Online Manuals CD or from the 3Com Web site.

For information on installing and operating two EMEs, see "Creating a Redundant Configuration" next in this chapter.

Hot Insert and Hot Swap

You do not need to turn off power to install modules in the Switch 4007 chassis. You can install modules while the chassis is operating. This action is called a *hot insert.*

You can also remove a module and install a replacement in that slot while the chassis is operating. This action is called a *hot swap*.

If your chassis is already operating in your network and you want to remove the primary (active) Management Module and retain uninterrupted chassis services, first ensure that a secondary (standby) Management Module resides in the chassis. When you remove the primary module, the secondary module activates quickly and seamlessly.

CAUTION: *Do not install or remove modules during the system power-on process or while software is downloading to any modules in the chassis.*

Installing Non-Management Modules

For a detailed installation procedure and other module-specific information, see the *Quick Start Guide* that accompanies the module. Guides for all modules that are supported in the Switch 4007 are available from the Switch 4007 Online Manuals CD or from the 3Com Web site.

Creating a Redundant Configuration

To establish a redundant management configuration, you must have two Management Modules (EMEs) installed in the chassis. The slots for management modules are obviously in the upper portion of the chassis due to their smaller size.

A redundant EME configuration provides a quick and seamless flow of management operation when the failover mechanism activates. There is no reboot of switching modules, no loss of data, and no interruption of service. You can also force a failover to occur from the EME CLI or the Web Management interface.

Installation

If you install two Management Modules, your system has redundancy in both management and system controller functions. 3Com recommends that you install the modules in the following order:

- Install the first EME in the lower-numbered slot (slot 8). This module will become the *primary* EME.
- Install the second EME in the other, higher-numbered slot (slot 9). This module will become the *secondary* EME.

Wait approximately two minutes before installing the second EME. If the EME that is installed in the higher-numbered slot boots up faster that the EME that is installed in the lower-numbered slot, then the EME in the higher slot will be the Primary. This also can happen if diagnostics are set to enable on the EME in the lower-numbered slot.

To determine how your system has assigned primary and secondary status to the two modules that you have installed, enter a `show module all` command.

For a detailed module installation procedure, see the *Enterprise Management Engine Quick Start Guide for the CoreBuilder 9000 Enterprise Switch.* This document is available on the Switch 4007 Online Manuals CD or from the 3Com Web site.

The Relationship Between Two Management Modules

The system categorizes the two EMEs as a primary and secondary management entities. The secondary module operates in *hot standby mode*, which means that it is constantly kept informed about the dynamic state of the management activities that are occurring on the primary EME.

Thus, the system generally treats both EMEs as a single logical device. (However, for some management activities, such as image download and Telnet connections, you must treat the two modules as separate devices.)

The primary EME and the secondary EME become synchronized after redundancy is established. When any configuration or non-volatile data is modified on the primary EME, the data is automatically modified on the secondary EME. New events are also stored on each EME, and the event log files are synchronized. All flash file system activity (copying files to, deleting files from) on the primary EME is mirrored on the secondary EME. Thus, if the primary EME fails for any reason, the secondary EME immediately takes over all primary functions.

Events that occurred before EME redundancy was established are not synchronized.

There are times when you may not want to automatically copy a file from the primary EME to the secondary EME (for example, temporary files). You can configure the system software to set these parameters during initialization. All files that are designated not to be automatically copied to the secondary EME are lost after a fail-over.

The Failover Process

If you remove (deinstall) the primary EME or if the module fails in some way, the following process occurs automatically:

1. The system initiates the fail-over mechanism (after, for example, the primary module fails or is removed).
2. The secondary EME becomes the primary EME.

   Because it learned all the configuration settings from the primary EME, it continues to provide all the management functions with no interruption to the system.
3. If you remove a failed EME that used to be primary and install a new EME in that slot, the new module remains secondary and automatically powers on in standby mode without modifying its configuration.

   If, after the failover occurs, the failed EME recovers to a normal operating condition, it will remain in the secondary state. The failover mechanism is non-revertive. Even if the problem that caused the failover is resolved, the the failover process does not switch the primary state back to the original primary EME.

The Standby LED, located on front panel of the EME that fails over, continues to display `Active`*. This is because the two components that make up the EME (SMC and SCC) are independent of each other. Therefore, when the SMC fails over, the SCC on the same EME continues to be active.*

Connectivity Rules

The following connectivity rules apply after you establish EME redundancy:

- You can only access the secondary EME through its console port or its auxiliary port (not the 10BaseT port).
- You cannot Telnet to the secondary EME from an external source because both the primary EME and the secondary EME share the same IP address for the front panel port. The front panel port is enabled only for the primary EME and disabled on the secondary EME. Therefore, when you attempt to Telnet to the shared IP address, you always access the primary EME.
- You cannot move from the primary EME to the secondary EME during a Telnet session (also known as *telnet-hopping*).

Verifying Management Module Operation

After you install a Management Module in the chassis and before you install other modules, verify that the Management Module is operating correctly. This section explains how to verify operation before you begin to enter commands.

To verify that your Management Module is operating correctly, watch the four-character LED display located on the front panel during the system power-on process. Table 5 shows the sequence of characters that appear in the LED display during a successful system power-on.

Table 5 The Management Module LED Display Readouts During Power-on

Characters in Display	Indication
random characters	Power-on has begun
none (blank display)	Power-on continues
`Diag`	Management Module is running self-diagnostic tests
`Cksm`	Management Module is calculating the checksum value
`Sec`	Management Module is in standby mode, if it is a Secondary
`Pri`	Management Module is active and ready, if it is a Primary

The system displays the following message when the Management Module is installed properly and you have made an RS-232 console connection:

```
CoreBuilder 9000 Enterprise Management Engine (vx.xx)
Copyright (c) 1999 3Com Corporation.
Login:
```

To ensure that a broken module LED is not providing a false indication of current conditions, enter the `show chassis` *command to verify that chassis operating conditions are normal.*

When two Management Modules are installed in a chassis, and the Primary Management Module fails over, the Standby LED on that Management Module continues to display `Active`*. This is because only the SMC component fails over while the SCC component remains active.*

The Display Button

The front panel of the Management Module includes a display button that is located next to the LED display, and labeled `DSPLY`. The LED display shows status information when you power on the Management Module, and shows `Pri` when the Management Module is running normally.

When the Management Module is running normally, the following information appears in the LED display when you press the display button:

- The first time that you press the display button, the LED display shows `FPEI` (Front Panel Ethernet Interface).
- The second time that you press the display button, the LED display shows `Vers` (Version) and then, after a few seconds, the release of software that is running the Management Module. Example: `0300`.

Making Management Connections

This section describes the connections that you can make to communicate with the Management Module. Choose the connection that is most appropriate to your installation. After you have connected to the Management Module, you can configure its characteristics.

Connecting to a 10BASE-T Ethernet Port

Connect the Management Module to a 10 Mbps (Megabits-per-second) Ethernet network or device using the front panel 10BASE-T Media Dependent Interface (MDI) port. The Management Module uses an RJ-45 connector for the 10BASE-T port.

Table 6 lists the 10BASE-T MDI port pinouts.

Table 6 10BASE-T (MDI) Port Pinouts

Pin	Signal Name
1	Transmit Data plus (TD+)
2	Transmit Data minus (TD-)
3	Receive Data plus (RD+)
4	no connection
5	no connection
6	Receive Data minus (RD-)
7	no connection
8	no connection

Using an MDI-to-MDI Crossover Cable

The 10BASE-T port is configured as an MDI or *host* port. To connect the Management Module to an MDI crossover (MDI-X) or *switch* port, use a standard RJ-45 jumper cable. To connect the Management Module directly to a host or another MDI port, use a crossover cable.

Table 7 lists the MDI-to-MDI crossover cable pinouts.

Table 7 MDI-to-MDI Crossover Cable Pinouts

Management Module Signal	Management Module Pin	Switch Pin	Switch Signal
TD+	1	3	RD+
TD-	2	6	RD-
RD+	3	1	TD+
RD-	6	2	TD-

Connecting to an RS-232 Console Port

Connect the Management Module to a terminal or modem using the RS-232 Console Port or RS-232 Auxiliary Port connectors. 9-pin connectors are used for the RS-232 ports.

Table 8 and Table 9 list the console port and auxiliary port pinouts.

Table 8 Console Port Pinouts

Pin	Signal Name
1	Carrier Detect (CD)
2	Receive Data (RD)
3	Transmit Data (TD)
4	Data Terminal Ready (DTR)
5	Signal Ground (GND)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	reserved

Table 9 Auxiliary Port Pinouts

Pin	Signal Name
1	Carrier Detect (CD)
2	Receive Data (RD)
3	Transmit Data (TD)
4	Data Terminal Ready (DTR)
5	Signal Ground (GND)
6	Data Set Ready (DSR)
7	Request to Send (RTS)
8	Clear to Send (CTS)
9	reserved

Table 10 lists 9-pin-to-9-pin assignments for connecting your PC to the front panel of the Management Module.

Table 10 RS-232 9-Pin-to-9-Pin Cable Connection Pin Assignments

Signal	Management Module Pin	DTE Pin	Signal
CD	1	N/A	Not Used
RX	2	3	TX
TX	3	2	RX
DTR	4	6	DSR
GND	5	5	GND
DSR	6	4	DTR
RTS	7	8	CTS
CTS	8	7	RTS
Reserved	9	N/A	Not Used

Table 11 lists 9-pin-to-25-pin assignments for connecting your PC to the front panel of the Management Module.

Table 11 RS-232 9-Pin-to-25-Pin Cable Connection Pin Assignments

Signal	Management Module Pin	DTE Pin	Signal
CD	1	N/A	Not Used
RD	2	2	TD
TD	3	3	RD
DTR	4	6	DSR

Table 11 RS-232 9-Pin-to-25-Pin Cable Connection Pin Assignments

Signal	Management Module Pin	DTE Pin	Signal
GND	5	7	GND
DSR	6	20	DTR
RTS	7	5	CTS
CTS	8	4	RTS
Reserved	9	N/A	Not Used

Using a Modem

The Management Module Console Port permits dial-in modem use. To use a dial-in modem:

1. Ensure that the modem supports the AT command set.
2. Select one of the following baud rates: 300, 1200, 2400, 4800, 9600, 19200, or 38400.

   The factory default is `9600`.
3. Place the modem in Dumb/Auto Answer mode. To do this, enter the commands that are listed in Table 12 from a terminal that is directly connected to the modem. Press Enter after each command.

Table 12 Modem Commands Required for Console Ports

a	at&F	Restore factory defaults
b*	at&d0	Ignore changes in DTR status
c	ats0=1	Auto-answer on first ring
d	ats0?	Verify auto-answer (should return 001)
e	atq1	Does not return result codes
f	ate0	Does not echo characters in command state
g	at&W	Save this configuration
h	at&Y	Define this configuration as default

* If you enter the `set terminal console hangup enable` command for modem use, you must change the DTR parameter as follows to ensure proper modem operation:

b	at&d2	Indicates hangup and assumes command state when an On to Off transition of DTR occurs

Verifying Network Connectivity

To verify that the chassis and all modules have been installed correctly:

1 Confirm that communication can be established on all network segments that you have enabled.

2 Confirm that the Network Activity LED on each installed module correctly indicates network traffic status. Table 13 lists the Network Activity LED status indicators.

Not all modules have Network Activity LEDs.

Table 13 Network Activity LED Status

10BASE-T Port Status	Network Activity LED Status
Link Down	Off
Receiving Traffic	Flashing Green
No Traffic (Port Enabled and Link Up)	Steady Green
Error	Steady Yellow

Troubleshooting Power-on Problems

Table 14 lists common problems that can arise when you install your EME and possible solutions. Under normal conditions, when you install the EME, the Status LED lights and the character display shows the EME's operating state.

Table 14 Power-on Troubleshooting

Symptom	Meaning	Corrective Action
Chassis power is on, but ACTIVE LED does not light	EME is in standby or has failed diagnostics	**1** Verify that the EME is installed correctly by following the installation instructions in Chapter 2. **2** Install the EME in the other slot in the chassis. **3** If the LED still does not light, the software on the EME may be corrupted. Try downloading a new copy of the software. (See Chapter 3.) If downloading software does not solve the problem, call your supplier for assistance.
Display reads `STBY`	EME is in standby mode.	**1** Wait 60 seconds to see if the EME corrects the situation itself. **2** If more than one EME exists in the chassis, verify that only one EME is set to Primary (using `show module all`). Then use the `reset eme` command to alleviate the problem. **3** Follow the corrective actions for the previous symptom.
Display reads `NRAM`	EME has failed diagnostics	Indicates a faulty non-volatile RAM device. Reinstall the EME or call your supplier for assistance.
Display reads `DRAM`	Indicates faulty on-board DRAM	Reinstall the EME or call your supplier for assistance.
Display reads `CARD`	Indicates faulty memory card	Reinstall the card or call your supplier for assistance.
Display shows numeric characters	EMC has failed diagnostics	Connect a terminal to the serial port and examine the diagnostic messages.

EME Technical Specifications

Table 15 lists general specifications, Table 16 lists power specifications, Table 17 lists environmental specifications, and Table 18 lists mechanical specifications for the EME.

Table 15 EME General Specifications

Element	Specification
Connectors	One front panel RS-232 shielded DB-9 connector for console port connections
	One front panel RS-232 shielded DB-9 connector for auxiliary port connections
	One RJ-45 10BASE-T Ethernet Port
Processors	One Motorola 68EC040 processor and two Motorola 68302 processors
Memory	16 MB of Flash EPROM 6 MB of RAM
	512 KB of Flash PROM for controller functions 512 KB of SRAM for controller functions
External Modem Support	For 100% Hayes-compatible modems
	Baud rates supported up to 38,400 baud

Table 16 EME Power Specifications

Element	Power Consumption
EME	12 W @ 5 V 1.0 W @ +12 V

Table 17 EME Environmental Specifications

Specification	Range
Operating temperature	0 °C to 50 °C (32 °F to 122 °F)
Humidity	Less than 95%, noncondensing
BTU/hr	46

Table 18 EME Mechanical Specifications

Element	Specification
Maximum slots	One EME per slot
Weight	0.45 kg (1 lb)
Dimensions	2.6 cm H x 20.8 cm W x 34.7 cm D (1.0 in. H x 8.2 in. W x 13.7 in. D)

4 CONFIGURING AND USING EME OPTIONS

This chapter describes how to configure an installed Management Module (EME). This chapter contains the following sections:

- Quick Reference Configuration
- Connecting to the System
- In-band Connections
- Configuring the Terminal
- Customizing Your System
- Configuring User Logins
- Configuring SNMP Values
- Configuring the Event Log
- Using the File System
- Resetting System Components
- Accessing the Administration Console
- Running Diagnostic Tests
- Obtaining Technical Assistance

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Quick Reference Configuration

Table 19 outlines the basic steps for configuring your Management Module (EME).

Table 19 Configuration Steps

Procedure*	Command
1 Configure your terminal to match the default EME communication settings.	See your terminal vendor's documentation

2 Configure EME terminal settings.
See "Configuring the Terminal" in this chapter.
	`set terminal console hangup`
	`set terminal console prompt`
	`set terminal timeout system`
	`set terminal timeout session`

3 Configure contact information, customize the prompt, and enable or disable diagnostics.
See "Customizing Your System" in this chapter.
	`set eme contact`
	`set eme diagnostics`
	`set eme location`
	`set eme name`

4 Set the time and date. See "Customizing Your System" in this chapter.	`set clock`

5 Configure user login criteria.
See "Configuring User Logins" in this chapter.
	`set login password`
	`set login administer`
	`show login`
	`clear login`

6 Set IP and SNMP parameters to enable network access and increase your management options.
7 See "Configuring SNMP Values" in this chapter.
	`set ip ip_address`
	`set ip subnet_mask`
	`set ip default_gateway`
	`set community`
	`set snmp`

* The order of configuration is important in some networks. Read the appropriate sections for more information.

Saving Configuration Values

When you make configuration changes to the EME using any of the `set` commands, they take effect immediately and they are saved permanently. Thus, do not make any configuration changes until you are fully aware of the consequences that these changes have on the system.

Connecting to the System

This section addresses procedures and commands that you can use to prepare the system (the EME's CLI) to receive connections from terminals or remote workstations or receive requests for access to the embedded Web server. This section also describes a feature that allows you to connect to another device from a local EME session.

Initial Access

When you first install a system, it does not have an IP address assigned to it. Thus, to get started, you must connect a terminal directly to the serial port on the Management Module (EME). When the system is powered on, the CLI automatically shows up on the terminal display.

Logging into the System

Before you can enter commands, you must log in to the system. To log in, enter your user name at the `Login:` prompt (factory default is `admin`) and your password at the `Password:` prompt (factory default is no password). Usernames and passwords are case sensitive.

After you log in with your user name and password, the system prompt appears. By factory default, the prompt appears as `CB9000>`. Enter commands at the prompt.Commands are not case-sensitive: you can mix uppercase and lowercase characters. For information about ways to enter commands, see "Entering Commands" next in this chapter.

Terminating a Connection

Whether you are connected in-band or out-of-band, when you no longer require a connection to the system, use the `logout` command to terminate the session:

```
CB9000> logout
Good-Bye
```

CAUTION: *If you set a timeout value for Telnet sessions (*`set terminal timeout session` *or* `set terminal timeout system`*) and it is reached, the EME terminates the session.*

Setting Up an IP Address for Telnet

If you configure an IP address, you have more options for accessing the system, such as Telnet and SNMP from remote workstations. To set up an IP address for Telnet purposes, follow these steps (for SNMP, you may need to configure additional options. See "Configuring SNMP Values" later in this chapter):

1. Verify that you have a direct terminal connection.
2. Log in to the system.

3 Use the `set ip ip_address` command to assign a unique IP address to the EME (RJ-45 type connector). Example:

CB9000> **set ip ip_address 195.36.58.27 ethernet_port**

4 Use the `set ip subnet_mask` command to assign a subnet mask to the EME.

For example, to set the subnet mask for a class B device, without subnetworks, enter a command similar to the following:

CB9000> **set ip subnet_mask 255.255.0.0 ethernet_port**

CAUTION: *The default subnet mask is ff.00.00.00. In networks in which the IP addresses begin with 151, the default subnet mask may conflict with internally reserved addresses. To avoid this situation, create the subnet mask before you set the corresponding IP address.*

5 Log out from your terminal session.

6 Connect your EME's 10BASE-T port to the network.

7 Use Telnet software to connect the system. Enter the system IP address in the appropriate field in the software interface.

8 Log in to the EME and manage the system as appropriate.

The EME supports up to four incoming Telnet sessions.

You cannot use Telnet to connect to an EME if the EME is in standby mode. You can connect an EME in standby mode to the network to provide redundancy only.

CAUTION: *Do not change the IP address of an EME that is already up and running from an in-band network connection. Doing so will terminate the session.*

Connecting to Remote Devices

The EME features a Telnet service for your convenience. To log in to another network device from an EME session, follow these steps:

1 Verify that you are logged in to the EME through a serial port.

You can enter the EME `telnet` *command only if you are logged in through a serial port (RS-232 port).*

2 Enter the telnet command and Specify the IP address of the remote device. Example:

CB9000> **telnet 192.34.67.101**

You can open a connection only with devices that support the Telnet protocol.

3 Log in to the remote device and manage the device using commands that are appropriate to that device.

You can create one outgoing Telnet session on each of the two console ports.

4 To log out of the remote device, use the appropriate command for that device.

After you have logged out of the second device, the local EME prompt reappears on the screen.

In-band Connections

To connect to the system in-band, you must connect the EME's Ethernet port to your network. This allows:

- In-band management using Telnet or SNMP.
- In-band download of operational boot code.

To connect to the EME in-band:

1 Make sure that you have set up IP connectivity for the single network that you plan to use for IP connectivity. See "Configuring User Logins"for more information.

2 Use Telnet or SNMP to reach the EME using the IP address that you assigned. In routed networks, you can connect to the EME using only the default gateway.

Serial Line Internet Protocol Connections

Vendors initiate Serial Line Internet Protocol (SLIP) sessions differently. Consult the documentation for your system. Although the 3Com SLIP implementation is as generic as possible, it may not function properly with SLIP implementations from other vendors.

To properly configure SLIP, follow these steps:

1 Assign an IP address and subnet mask to the SLIP interface on the EME using the following command:

set ip subnet_mask `<mask>` **serial_port**

where `<mask>` is the workstation's IP subnetwork

Assign the same IP address and subnet mask to the serial port of the device on the other end using the following command:

set ip ip_address `<address>` **serial_port**

where `<address>` is the EME IP address.

2 Assign the terminal settings using the following command:

`set terminal <port> <baud, data_bits, hangup, mode, parity, stop_bits, terminal_type>`

Set the terminal to match the remote terminal settings.

3 Enable SLIP mode using the following command:

`set terminal` `<port>` **`mode slip`** `<address>`

where `<port>` is auxiliary or console, and `<address>` is the workstation's IP address.

To end the SLIP session do one of the following:

- From a remote SLIP connection, send a break character to the EME
- From the EME command line interface, set the SLIP port to command mode or disable the interface.

Configuring Access to the Web Interface

To enable or disable access to the embedded Web management interface, use the following commands at the prompt:

```
set web access disable
set web access enable
```

To set a time value for the Web interface session to time out, use the following command at the prompt:

```
set web timeout system <time period>
```

The available time span is 0 - 30 minutes with 0 indicating there is no timeout.

You can view the current settings on your system for Web Access and Web Timeout through the following EME commands:

```
show web access
show web timeout
```

Entering Commands

This section describes ways to enter commands and display command options.

The Command Completion Feature

The command completion feature allows the interface to accept abbreviated command input. You need only to enter a minimum number of characters to distinguish the command from other acceptable choices and then press the spacebar to complete the command.

Procedure:

1 Enter a command (for example, the `show` command).

2 Enter the first several letters of the selected command parameter.

3 Press the spacebar to complete the command.

4 Press Enter to process the completed command.

Example:

```
CB9000> sh [spacebar]
CB9000> show
CB9000> show cha
CB9000> show cha [spacebar]
CB9000> show chassis
CB9000> show chassis [Enter]
```

If the characters that you enter are not sufficient to determine a unique command, the EME waits for you to enter more characters. For example, entering the letter “s” and pressing the spacebar is not sufficient for the EME to determine which command to issue because other commands also start with the letter “s” (that is, `set` and `show`).

Listing Command Options

To display a list of top-level menu options from the CB9000> prompt, type **?** and press Enter. Then, to display a list of submenus or options for any of those options, type one of the options followed by one space and **?** and press Enter.

The question mark does not appear when you type it. It is included in examples for illustration purposes only. You must include a space between the command and the **?***.*

After the system provides the list requested, it presents the prompt with the last command that you entered, minus the **?**.

Example:

```
CB9000>> ?
Possible completions:
        clear
        connect
        download
        logout
        ping
        reset
        servdiag
        set
        show
        telnet
        upload

CB9000>>
```

You can continue using **?** in this manner. Example:

```
CB9000>> show ?
Possible completions:
        chassis
        clock
        community
        eme
        event_log
        file
        host
        interface
        inventory
        ip
        login
        module
        power
        security
        servdiag
        snapshot
        snmp
        sntp
        terminal
        web

CB9000>> show
```

Each command as a list of options associated with it. The options that are available to complete the command may depend on the type of module that is in the chassis slot.

If you enter an option from the list followed by a **?**, any additional options that can be used appear or, if none are available, the following message appears: `Confirm with Carriage Return.`

Keystroke Functions

You can alter your keyboard input using specific keyboard functions and control sequences. If you press Enter in the middle of a command entry when a parameter is expected, the EME prompts you for additional information.

Table 20 lists these keystrokes and their functions.

Table 20 Terminal Keystroke Functions

Keystroke	Function
Backspace	Moves the cursor back one character and deletes that character
Ctrl+C	Terminates the current command and returns to a blank command line at any time
Ctrl+D	Closes a Telnet session
Ctrl+R	Retypes the previous command string on the command line
Delete	Moves the cursor back one character and deletes that character
Enter	Implements the command
spacebar	Completes an abbreviated command
?	Displays the available command options

Configuring the Terminal

This section describes:

- Configuring the Terminal to Default Settings
- Changing the Terminal Configuration
- Customizing Terminal Settings

Configuring the Terminal to Default Settings

Configure the terminal that is attached to the serial port on the EME to the same parameter settings as the EME. In doing this, you allow the terminal and EME to communicate. Initially, the terminal settings must match the factory-default settings of the EME, as specified in Table 21. To display the current terminal settings, use the `show terminal` command. To access the Administration Console, use the `connect <slot>.1` command.

Table 21 Terminal Defaults and EME Options

Parameter (SET TERMINAL +)	EME Options (when connected)	Factory Default
Baud	300, 1200, 2400, 4800, 9600, 19200, 38400	9600
Data_bits	7 or 8	8
Parity	odd, even, or none	none
Stop_bits	1 or 2	1
Hang_up	enable or disable	disable
Mode	command line or slip	command line
Terminal_type		VT100

To configure the terminal:

1. Consult the user guide that was shipped with your terminal for instructions about setting the terminal values.
2. After you configure your terminal to match the factory defaults of the EME, press Enter.

   The following message appears:

   ```
   CoreBuilder 9000 Enterprise Management Engine (vx.xx)
   Copyright (c) 1999 3Com Corporation
   ```
3. At the `Login` prompt, enter a login name.

   The default login name is `admin`

   The EME prompts you for a password. By default, there is no password.

4 Press Enter. The EME displays the following message and prompt:

```
Welcome to Administer service on CB9000.
CB9000>
```

You are now logged in as the `admin` with full access to all commands.To show the current terminal settings, use the `show terminal` command.

After terminal settings are complete, you can configure the newly installed EME, and all other modules in the chassis.

Changing the Terminal Configuration

To change the terminal configuration, use the `set terminal` command using the EME options listed in Table 21. The syntax for the command is:

```
set terminal <port> <option>
```

Where `<port>` is either `console` or `auxiliary` and the options are as listed in Table 21.

After you enter each new `set terminal` command (changing the baud rate, for example), you must change the settings for the terminal to match the new setting before you can reestablish communication.

Customizing Terminal Settings

The EME allows you to change the following optional terminal management settings to customize your terminal connection:

- Terminal hangup
- Terminal prompt
- Terminal timeout value
- Terminal type

These terminal settings are optional, and apply to both terminal ports. Any changes that you make to the terminal parameters are automatically saved when you press Enter.

Setting Terminal Hangup

If you use a modem connection to log in to the EME, use the `set terminal console hangup` command. This command causes the EME to de-assert the RS-232 DTR signal when you log out of the EME. This forces the modem to hang up the connection and may help prevent unauthorized access.

The default for the `set terminal console hangup` command is `disable`. When the command is set to `disable`, the DTR signal remains asserted when you log out.

Example:

```
CB9000> set terminal console hangup enable
```

Setting Terminal Prompt

Use the `set terminal prompt` command to customize the terminal prompt for each EME. Use this prompt to identify the EME that you are connected to when logged in to a remote EME. The default is `CB9000>`.

To customize your terminal prompt, use the `set terminal prompt` command. Example:

```
CB9000> set terminal prompt EME3>
```

To avoid confusion, use the same identification for both the terminal prompt and for the name of your EME.

Setting Terminal Timeout Value

Use the `set terminal timeout session` or `set terminal timeout system` commands to specify the amount of time that you want your terminal to remain active during the absence of any keyboard activity. The `session` keyword applies the timeout value to the current terminal session, and the `system` keyword applies the timeout value to all sessions on the system.

Use this feature to keep unauthorized users off of the system if you leave your terminal without logging out. The default for the command is `0`, which means that no timeout has been set and the terminal cannot be logged out automatically.

To set the timeout period (value expressed in minutes), use the `set terminal timeout` command. You can specify up to 30 minutes.

Example:

```
CB9000> set terminal timeout system 10
```

After you set the timeout, the terminal automatically logs you out of the system if there is no terminal (keyboard) activity for the period of time that you have specified. In this example, logout occurs after 10 minutes of keyboard inactivity.

Setting Terminal Type

Use the `set terminal console terminal_type` command to define a terminal type for use with outbound Telnet sessions. The system sends the terminal type to the device that is connected to the EME when initiating the Telnet session. The terminal type setting enables the device to send the proper control sequences to the EME, which appear on the EME terminal.

The following command defines the terminal type as a VT100 terminal on the console port:

```
CB9000> set terminal console terminal_type vt100
Terminal type changed.
```

Troubleshooting the Terminal Interface

Table 22 lists some common problems that can occur as you configure the EME to communicate with a terminal.

Table 22 EME Terminal Interface Problems

Symptom	Corrective Action
Nothing appears on the screen (screen is blank).	■ Make sure that the RS-232 cable meets the specifications in Chapter 2. ■ Make sure that the RS-232 cable is securely connected to both devices. ■ Verify that the baud rates match for the terminal and the EME. (See Chapter 4)
Garbled characters appear on the screen.	Verify that the EME and the terminal settings match for baud, data bits, stop bits, and parity. The default baud rate is 9600, 8 data bits, no parity, 2 stop bits.
The `set` command does not work.	Make sure that you are logged in as `admin` and that you are connected to the primary EME (display shows `Rdy`).
You use abbreviated input, but pressing the spacebar does not complete the command.	Enter enough characters for the EME to distinguish between different commands and options. Enter **?** for a list of available options.

Table 22 EME Terminal Interface Problems (continued)

Symptom	Corrective Action
Characters are lost when connected to the EME through a modem.	Make sure the STOP_BITS value on the terminal is set to 2STOP_BITS.
The management prompt on the screen is not as you set it.	You may be connected to a remote device. See the `telnet` and `logout` commands described in Chapter 6.
You do not receive any statistics from the chassis.	Make sure that you have properly configured EME IP information.
	Make sure that the statistics groups are enabled.
The `>>` prompt appears on the screen.	The EME is running in maintenance mode. Enter **`boot`** to return to management mode and the `CB9000>` prompt.
Module fails to respond after download.	Retry the download. If the module appears not to be operating, contact your service provider.
Module reports that a particular subnetwork is reserved.	Subnet 151.104.252.0 is reserved for chassis use. Use a different subnetwork.
Statistics are inaccurate.	EME statistics are designed to identify problems on the network. They may not be 100% accurate.
Module appears after `show inventory` command, but not after the `show module` command.	Reset the EME and retry the command.
	Replace the EME.

Customizing Your System

You can alter the factory defaults to customize various aspects of your system, including its name, as associated contact name, and whether the EME runs diagnostics as part of its boot sequence.

Assigning a Unique Name

You can assign a unique name to an EME. Subsequently, you can use this name instead of the IP address or MAC address to reference the EME (for example, when using Telnet).

To assign a unique name (up to 31 characters) to your EME, use the `set eme name` command. For example, to set the EME name to `bldg3floor2`, enter:

```
CB9000> set eme name bldg3floor2
```

Use the same identification to specify the terminal prompt and the name for your EME.

To display the current system name, use the `show eme` command.

Setting EME Diagnostics

You can set the EME to bypass diagnostics. When you reset the EME (or reboot it) with diagnostics enabled, the EME performs diagnostics before it returns to full functionality. The EME boots faster with these diagnostics disabled. Diagnostics are enabled by default.

To prevent the diagnostic sequence from being part of the boot process, enter:

```
CB9000> set eme diagnostics disable
```

Assigning a Contact Name and Location

The EME can store the name of a service contact and chassis location for reference. Use the `show eme` command to display the current contact name and location of the EME.

To identify the location of the EME and the name of the person responsible for the EME, use the following commands:

```
CB9000> set eme location
CB9000> set eme contact
CB9000> set eme name
```

After you enter each command, the EME prompts you to enter a line of text, which can be up to 78 characters:

```
CB9000> enter one line of text:
```

The EME commands time out if you do not enter text within 15 seconds.

Configuring the Internal Clock

Use the `set clock date_time` command when you install the EME into your chassis to establish a starting time, date, and day. Define this setting only once, because changing the clock setting may affect real-time statistics gathering. To display the current time, use the `show clock date_time` command.

To set the 24-hour internal clock to 5:58 PM, Tuesday, May 9th, 2000, enter:

```
CB9000> set clock date_time 17:58T00/05/09 tuesday
```

The internal clock is powered by its own battery and continues to work even if the chassis loses power. Even when the EME is powered off, this battery is designed to operate for 10 years. You can change the timezone using the `set clock time_zone` command. You can also enable your chassis for daylight savings time using the `set clock daylight_saving_time` command.

The `set clock time_zone` command:

- Allows you to set local time zone and daylight savings time values.
- Displays the following time zone table:

```
Index    Time Zone
          1 [GMT+0:00] GMT/WET/UT
          2 [GMT-1:00] WAT
          3 [GMT-2:00] AT
          4 [GMT-3:00] Brasilia/Buenos Ar/GeorgeTown
          5 [GMT-4:00] AST
          6 [GMT-5:00] EST
          7 [GMT-6:00] CST
          8 [GMT-7:00] MST
          9 [GMT-8:00] PST
         10 [GMT-9:00] YST
         11 [GMT-10:00] AHST/CAT/HST
         12 [GMT-11:00] NT
         13 [GMT-12:00] IDLW
         14 [GMT+1:00] CET/FWT/MET/MEWT/SWT
```

```
    15 [GMT+2:00] EET
    16 [GMT+3:00] BT
    17 [GMT+4:00] ZP4
    18 [GMT+5:00] ZP5
    19 [GMT+5:30] Bombay/Calcutta/Madras/New
                 Dehli/Colombo
    20 [GMT+6:00] ZP6
    21 [GMT+7:00] WAST
  22 [GMT+8:00] CCT
  23 [GMT+9:00] JST
  24 [GMT+9:30] Darwin/Adelaide
  25 [GMT+10:00] EAST/GST
  26 [GMT+11:00] Magadan/Solomon Is/N. Caledonia
  27 [GMT+12:00] IDLE/NZST/NZT
  28 Input an offset from GMT
Select timezone index {1-28|?} [1]:
```

- - Adjusts the server reply universal time to local time properly.

The default time zone is Greenwich Mean Time (GMT).

Configuring User Logins

This section describes the different commands for assigning user (network personnel) access levels and login functions:, setting up passwords, and adding or deleting user profiles.

User Access Levels

The EME provides three levels of user access:

- **Administer** — The user can perform all tasks, reset and configure modules, and add and change passwords, as well as:
  - Configure EME IP address information
  - Configure community tables
  - Download new operational and boot code
- **Write** — The user can perform most commands except those that configure IP information, community tables, and download software.
- **Read** — The user can display information about network configuration and operation (except community table information). Read users can change their own passwords with a `set` command.

To add login names, you must be logged in with a user name that has been assigned Administer access.

User Login Functions

You can configure up to 10 user logins in any combination of access levels using the EME. More than one user at a time can log in to the command interface.

Login Limitations

Only one user at a time can log in with Administer privileges. If a second user with Administer access privileges tries to log in, that user has access to Write-level functions only. Up to four remote (Telnet) sessions can be established at one time.

Administer Access

Because the EME allows only one user with Administer privileges to log in at a time, the software includes a special `set login administer` command. If an Administer user logs in and is granted only Write privileges, that user can use the `set login administer` command with the following implications:

- The current Administer user is logged out of the EME.
- The user who enters the command immediately assumes Administer privileges.

Setting the Password

By default, the EME has no password. The first time that you log in, you press Enter at the `Password:` prompt. To set a password for the default log in username, use the `set login password` command.

Setting a password for the default login name makes the system more secure. Without a password, other users can log in with Administer access and change system configuration settings.

Choose a password that you can remember. If you forget or lose the Administer password, you cannot log in and perform Administer-level functions. See "Resetting the EME to Default Values" on page 105 for information about how to recover system defaults if you forget or lose the Administer password.

Example:

1 Enter the `set login password` command.

```
CB9000> set login password
```

2 Enter the password at the prompt.

```
Enter Login password:
```

3 At the next prompt, reenter the password.

```
Verify - re-enter password:
Login successfully entered.
```

Adding New Users

You can configure up to 10 user logins, with access rights as described previously in this chapter.

To add a new user, you must have Administer access.

To add a new user:

1 Log in using an Administer name and password.

2 At the prompt, enter:

```
set login administer <login type>
```

Where `<login type>` is the type of user that you are adding: `administer`, `write`, or `read`

The system prompts you for your password (to confirm your right to set new passwords) as follows:

```
Enter current session password for user “admin”:{enter
password}
```

3 Enter your password.

4 At the `Enter Login Name:` prompt, enter the login name for the user that you want to add.

5 At the `Enter Login Password:` prompt, enter the user's login password.

6 At the `Verify - re-enter password:` prompt, enter the new password again.

The system acknowledges the new password by displaying:

`Login successfully entered.`

Showing Current Users

To show the existing login names for the EME, enter:

`CB9000>` **`show login`**

The following type of information appears:

```
Login Table:

      Index   Login Name        Access          Active Sessions
      -----   ---------------   --------------  ---------------
          1   admin             Administer           1
          2   Pete              Write                0
          3   Larry             Write                0
          4   Marie             Administer           0
          5   Richard           Read                 0
          6   [not used]
          7   [not used]
          8   [not used]
          9   [not used]
         10   [not used]

Active Login Sessions:

     Login Name       Session Type              Session Time
     ----------       ------------              ------------
     admin            Local Administer          0 days 00:28:43
```

Table 23 describes the fields in the `show login` display.

Table 23 Fields in the `show login` Display

Column	Description
Index	Index number of each of the 10 available logins
Login Name	Name assigned to each login
Access	Privilege level assigned to this login name (Administer, Write, or Read)
Active Sessions	Number of active sessions under this login name
Active Login Sessions	Session Type — User privileges and whether session is local or remote Session Time — Length of the session

Clearing Login Names

You may want to clear login names from the EME periodically to help ensure system security. Only a user with Administer access can clear other users.

You can enter either the index number of the user or users that you want to clear or `all` to clear all users except yourself (as the Administer user). Use the `show login` command to display all login names and their corresponding index numbers. If you clear all users, you can log in to the EME with the default username (`admin`) and no password.

To clear the username with index number 3, enter:

```
CB9000> clear login 3
```

To clear all users, enter:

```
CB9000> clear login all
```

Configuring SNMP Values

The Simple Network Management Protocol (SNMP) is a standard that is defined by the Internet Engineering Task Force (IETF). SNMP information is encapsulated in a UDP and IP packet, which in turn, is encapsulated in an appropriate protocol-specific frame.

This section describes the configurable options on the Management Module (EME) that relate to SNMP management.

The Administration Consoles of individual switching modules also offer SNMP commands that you may need to configure.

Interaction Between the EME and SNMP

The EME interacts with SNMP to:

- Respond to SNMP requests.
- Generate SNMP traps.
- Act as an agent in an SNMP-managed environment, enabling you to configure your EME.

If you plan to manage your chassis using an SNMP workstation, you must enable the 10BASE-T front panel Ethernet port and set the following attributes for the EME:

- IP connectivity
  - Subnet mask
  - IP address
  - Default gateway
- Community table
- Alerts (optional)
- Trap receive

Setting Up IP Connectivity

Assigning an IP Address to the EME

Use the `set ip ip_address` command to assign a unique IP address to the EME.

Example:

```
CB9000> set ip ip_address 195.36.58.27 ethernet_port
```

Setting a Subnet Mask

Use the `set ip subnet_mask` command to assign a subnet mask to the EME.

For example, to set the subnet mask for a class B device, without subnetworks, enter a command similar to the following:

CB9000> **set ip subnet_mask 255.255.0.0 ethernet_port**

CAUTION: *The default subnet mask is ff.00.00.00. In networks in which the IP addresses begin with 151, the default subnet mask may conflict with internally reserved addresses. To avoid this situation, always set a subnet mask before you set the corresponding IP address.*

CAUTION: *Do not change the IP address of an EME that is already up and running from a network connection. Doing so terminates the session.*

Defining a Default Gateway

Use the `set ip default_gateway` command to assign default gateways to networks. The default gateway is the IP address of the gateway (for example, a router) that receives and forwards packets whose addresses are unknown to the local network. The EME uses the default gateway when sending alert packets to a management workstation on a network other than the local network.

For example, to specify that the gateway with address 195.36.58.1 is the default gateway, use the following command:

CB9000> **set ip default_gateway 195.36.58.1 ethernet_port**

You must connect the front panel Ethernet port to the locally attached network.

Showing and Clearing IP Settings

Use the `show ip` or `show interface` commands to view IP parameter settings. Use the `clear ip <index #>` command to clear parameter settings. Before you clear an IP interface, use the `set interface <index#> disable` command to ensure that the interface is not in use.

Creating a Community Table

Use the Community Table to define:

- SNMP stations on the network that access information from the EME
- SNMP stations that receive traps from the EME

To enable the EME to receive SNMP alarms, you must add the following items to the community table of the SNMP device that generates the alarms:

- The EME IP address
- Accompanying attributes

The EME Community Table can contain up to 10 IP community entries. You may assign one of the following attributes to the IP addresses:

- **Read-only** — Allows the specified IP address community to read SNMP objects using the SNMP `get` and `get next` commands.
- **Read-write** — Allows the specified IP address community to read and write SNMP objects using the SNMP `get, get next,` and `set` commands, respectively.
- **Trap** — Sends a trap to the specified IP address when an event occurs.
- **Read-trap** — Allows the specified IP address to read SNMP objects and receive traps.
- **All (read-write and trap)** — Allows the specified IP address to read SNMP objects, change the objects using the SNMP `set` command, and receive traps.

Use the `set community` command to create a Community Table entry. For example, to add a community name of NCS with the IP address 195.36.58.217 that has read_write access, enter the following command:

```
CB9000> set community NCS read_write 195.36.58.217
```

Community entry names are case-sensitive. For example, NCS and ncs are different community names. You can use the `show community` *command to view existing community entries.*

The wildcard value of All appears as `***.***.***.***` *for IP addresses. The value for All access privileges is only* `all`*.*

Configuring a Trap Destination

To set up a destination for SNMP traps, use this command at the EME prompt:

```
set snmp trap destination <community name> <ip address>
```

Where `<community>` is the community string of the selected trap where you want to send the trap and `<number>` is the IP address of the trap receiver.

Configuring the Authentication Alert Setting

To enable or disable the feature that sends an alert to the management workstation when someone tries to gain access to the EME and the IP address or community name is not valid for the attempted read or write operation, use following commands at the prompt:

```
set snmp trap filter authentication disable
set snmp trap filter authentication enable
```

Configuring Trap Options

As a trap receiver, the EME receives traps from other SNMP devices, including switching modules in the chassis, that have the EME IP address in their Community Table. For example, to allow an EME to function as the trap receiver for other SNMP devices on the network, use the following command:

```
CB9000> set snmp trap receive enable
```

To enable that device to send traps to the EME, add the EME IP address to that device's Community Table.

To disable the ability of the EME to receive SNMP traps, use:

```
CB9000> set snmp trap receive disable
```

You can also configure the following EME trap characteristics:

- Alert transmission — You can configure the EME to transmit an SNMP trap to trap receivers that you define, and you can specify the system events that trigger these alerts.
- Trap receivers — You can specify the network management stations that receive alerts from the EME. To do so, you must create a new community table entry for each network management station.

SNMP traps are transmitted only through an in-band IP routing interface.

To enable or disable the trap filter link state, use these commands:

```
set snmp trap filter link_state disable
set snmp trap filter link_state enable
```

To enable or disable the trap filter for detecting a Spanning Tree Protocol topology change on one of the switching modules, use these commands:

```
set snmp trap filter topology_change disable
set snmp trap filter topology_change enable
```

A topology change trap is sent by a bridge. Only the topology change trap is filtered with this command. You will continue to see any new root traps.

Viewing SNMP Extensions and Traps

To view the SNMP extensions on your system, use this command at the EME prompt:

```
show snmp extensions
```

To view SNMP traps on your system, use this command at the EME prompt:

```
show snmp traps
```

Interpreting EME Trap Messages

The EME console receives a trap message when a change is made or an error occurs in a chassis that has an installed EME. The designated trap receiver (for example, a management workstation) also receives a trap if you have entered this information in the EME community table.

For example, if you remove a module from a chassis, the EME sends messages that describe the change to the console:

```
Message received from this device on 15:58 Fri 09 Jul 99:
Enterprise:   3Com
Enterprise Specific Trap:    Module Down
Message Information:
Slot Number:   6
Subslot Number:   1
Module Type Number:   6
Module Description:
Message received from this device on 15:58 Wed 21 Jul 99:
Enterprise:   3Com
Enterprise Specific Trap:   Slot Down
Message Information:
Slot Number:   6
```

Table 24 describes the first two fields in the trap message. The remainder of the fields are dependent upon the type of trap that is received and are self-explanatory.

Table 24 EME Trap Message Fields

Field	Description
Enterprise	Describes the enterprise (organization) responsible for this type of trap message.

Enterprise-Specific Trap	One of the following trap messages:
	`Module Up` or `Module Down`
	`Slot Up` or `Slot Down`
	`Port Up` or `Port Down`
	`Trunk Up` or `Trunk Down`
	`Fatal Error`
	`Environment Change`

SNMP traps are sent to the EME console when traps occur. An example of an SNMP trap is when a device attempts to gather information (read) from the EME, but the address of the device was not added to the community table with that access level. The message that appears in this instance is similar to the following example:

```
Message received from this device on 15:58 Fri 09 Jul 99:
Enterprise:   3Com
SNMP Generic Trap:    SNMP Authentication Failure
Message Information:
 Authentication Failure Address: 192.104.6.163
```

Obtaining More Information About SNMP

More information about protocols is available from the references in Table 25.

Table 25 Protocols and References

Protocol	Reference
UDP	RFC-768
SNMP	RFC-1157
IP	RFC-791
Telnet	RFC-854
ARP	RFC-826
802.2	ISO/DIS 802/2

Configuring the Event Log

The EME maintains a log of informational events, nonfatal errors, and fatal errors that occur on all modules in the chassis. Event log entries are stored in the chronological order in which they are received.

When two EMEs are installed in the chassis, only the Primary EME collects information. Each EME only stores events that occur while that EME is the Primary.

You can configure the following event log characteristics:

- Amount of memory allocated to storing events

  Event log memory is allocated in 64k blocks. The default allocation (also the minimum setting) is eight 64k blocks (0.5 MB). The maximum setting depends upon available memory.

- Action for the EME to take when the event log buffer is full

  You can set the system to stop logging events, or to begin overwriting old events.

- Mechanism that triggers the EME to copy the event log to a file server

  The EME can upload the event log when the event log reaches a certain percent usage (default is 80 percent), when a user-defined time interval has passed, or when you initiate the event log upload.

  Before the event log can be copied to a file server, you must create a public-access file on the file server to which the event log is written.

  Uploaded event logs are larger than the size that appears on the EME because they are stored in a compressed format.

To display the status of the event log's characteristics, use the `show event_log status` command.

To display the contents of the Event Log, use the `show event_log unfiltered` command with the `nonverbose`, `verbose`, or `<number>` options.

The event log full default is `WRAP`.

Using the File System

The file system is an area on the Management Module that stores:

- Software configuration files
- The event log

The file system also acts as a temporary storage area for software images that are downloading through the Management Module to other modules in the chassis.

Under most conditions, you do not need to access or manage the file system. The file system supports commands that allow you to view the files in the file system or to delete certain files.

Software Configuration Files

The following commands are available to display and manage the file systems, which store software configuration files:

- `show file`
- `clear file`
- `clear file_system`

Displaying Files in the File System

The `show file` command displays files in the file system storage. Example:

```
CB9000> show file

Eme flash disk directory contents list:
Current number of files is: 13
Maximum number of files was: 15

FileSize      Date          Time      FileName
-------   -----------      -----      --------
170551     Jul 25 1999     10:27:26   EventLog
71288      Jul 24 1999     10:39:01   BladeConfig.08.01
71288      Jul 24 1999     10:39:10   BladeConfig.09.01
82904      Jul 25 1999      9:49:15   BladeConfig.10.01
82904      Jul 25 1999      9:49:19   BladeConfig.11.01
82904      Jul 25 1999      9:49:23   BladeConfig.16.01
82904      Jul 25 1999      9:49:27   BladeConfig.07.01
82904      Jul 25 1999      9:50:00   BladeConfig.14.01
82904      Jul 25 1999      9:50:04   BladeConfig.13.01
82904      Jul 25 1999      9:50:08   BladeConfig.12.01
82904      Jul 25 1999      9:50:12   BladeConfig.15.01
```

```
82904      Jul 25 1999      9:52:15   BladeConfig.06.01
82904      Jul 25 1998      9:52:19   BladeConfig.05.01

Number of files:                         13
Number of bytes in file system:          11945984
Number of bytes used:                    1157120
Number of bytes available:               10788864
Number of bytes cleaned:                 10788864
```

Deleting Specified Files From the File System

The `clear file <filename>` and `clear file all` commands delete the files that you specify from the file system. If you enter `all` to delete all files, the system prompts you to confirm that you want to delete each file before you delete it. If you enter `n` (no), the system does not delete the file. Example:

```
CB9000> clear file all

Are you sure you want to delete file EventLog?(y/n): n
Are you sure you want to delete file a.1 (y/n): y
File a.1 deleted.
Are you sure you want to delete file a.11? (y/n): y
File a.11 deleted.
```

The system continues to prompt you about files until all files are either deleted or saved.

Deleting All Files and Resetting the Management Module

The `clear file_system` command deletes all files that are stored in the file system, reinitializes the file system, and resets the Management Module. This command clears the file system even if files are currently open or being updated.

Use this command only if the combined number of bytes that individual files use is inconsistent with the figure in the Number of bytes used: field in the `show file` display. The number of bytes that each file uses appears in the FileSize column of this display.

CAUTION: *This command deletes all files from the file system and reinitializes the file system storage area. Do not use this command unless the file system has been corrupted in some way.*

Before you clear the file system, you can store a copy of the event log on the file server with the `upload eme event_log <ip address> <filename>` command, if you want to save the content of this file. Example:

```
CB9000> clear file_system
        !!WARNING!!
This command will clear all files and reset the EME.
Consult the user guide for information on operational
considerations before continuing with this command.

Do you wish to continue with clear file_system
command?(y/n):y
Preparing to clear file_system.
Ready to clear file_system.
Do you wish to continue with clear file_system
command?(y/n):y
Clearing file system please wait for EME to reset.
```

After the reset is complete, you can log in to the EME.

Resetting System Components

Certain situations require that you reset power to the entire chassis, certain switching modules, or the EME. This section describes the commands for performing such actions.

Resetting the Chassis

Use the `reset chassis` command to reboot all of the installed modules and the chassis itself, including the EME.

This command performs a hardware reset of the chassis and all installed modules. Diagnostic routines execute (if enabled) and traffic forwarding may be briefly interrupted. After the chassis reset is complete, you must log back in to the primary EME before you can enter any other commands.

Resetting Switching Modules

Use the following commands to reset modules installed in the chassis from the EME prompt:

- `reset module <slot>.<sublsot> cold` — Use this command after you downgrade software releases. This command cycles power (off/on) to the indicated module and runs its diagnostics (Diags) software which updates the module's Power On Verification (POV) software.
- `reset module all cold` — Use this command when you downgrade from Release 3.0.0 software to v2.1. This command performs a power cycle on all of the modules installed in the chassis and runs the Diags, which updates all of the modules' Power On Verification (POV).
- `reset module <slot>.<sublsot> warm` — Use this command after you download a configuration file to a module. This command is the same as the `reset module <slot>.<sublsot>` command.
- `reset module all warm` — Use this command after you download configuration files to all of the modules installed in the chassis. This command is the same as the `reset module all` command.

Resetting the EME

Use the following commands to reset the EME:

- `reset eme cold` — Use this command after you downgrade software releases. This command cycles the power (off/on) and runs the EME diagnostic (Diags) software which updates the EME's Power On Verification (POV) software.
- `reset eme warm` — Use this command after you download a configuration file to an EME. This command is the same as the `reset eme` command.

Resetting the EME to Default Values

You can reset the EME to its user-configurable values and options to their default values using the `FORCE` command. The `FORCE` command resets all EME. If you have forgotten or lost the Administer password, this command is the only way to reset this password to the default value, which is no password. You cannot use this command remotely because you must press the EME Reset button on the front panel after you enter the command.

CAUTION: *Do not use this command unless absolutely necessary. This command resets all user-configurable values and options to defaults, and terminates all network communications. You will need to reenter all values and options that you changed with* `set` *commands.*

Choose an Administer password that you can remember, so that you do not have to use the `FORCE` *command.*

The `FORCE` command is case-sensitive. Use all uppercase letters.

To reset the EME to factory defaults, follow these steps:

1. Log out of the EME using the `logout` command.
2. Press Enter for the `Login:` prompt.
3. Enter **`FORCE`** as the username.

   The system prompts you for a password.

4 Enter **FORCE** as the password.

5 Press the EME Reset button *within 5 seconds* after you have pressed Enter. A series of reports appear ending with the following message:

```
NVRAM not initialized or corrupt. Loading factory defaults.
```

You can now log in to the EME using default values. (At the `Login:` prompt, enter **admin**, and at the `Password:` prompt, press Enter.)

After you perform the `FORCE` *operation, the EME that was previously configured as the Secondary EME becomes the Primary EME.*

Accessing the Administration Console

The software in switch fabric modules and switching modules includes a menu-driven command line management interface, called the *Administration Console*. To access a module's Administration Console, use the `connect` command from the EME and specify the modules's slot number and subslot number (which is always 1). For example, to access a module in slot 4, enter:

```
CB9000> connect 4.1
```

When the Administration Console appears, you can enter options from the top-level menu at the module prompt. These options lead to other menu options.

To exit the module and return to the EME management console, enter the `disconnect` command at the module prompt.

See the *Switch 4007 Command Reference Guide* for more information about the commands that are available through the Administration Console. Also see the chapters in Part III of this guide for extended information about those features.

Running Diagnostic Tests

Use the `servdiag` command to run diagnostic tests on any switching module that you specify. This command is useful if you suspect a problem on the module or if you notice that the module is behaving inconsistently. The syntax for this command is:

```
servdiag <slot.subslot> <test>
```

Two types of tests are available:

- **Boot** — This test is the same module diagnostic test that runs automatically when you power on a module. This test takes up to 4 minutes to run.
- **Extended** — This is a series of tests that include the Boot test and a series of subtests. This test takes up to 15 minutes to run.

While the module runs these diagnostic tests, it does not pass network traffic. Do not use this command unless you suspect a problem on the module, and you do not need to use the module in your network.

The following example runs the Boot test on an interface module in slot 2. This module passes the test.

```
CB9000> servdiag 2.1 boot
Test may take up to 4 minutes and 0 seconds.
Do you wish to continue (y/n): y

Module 02.01 accepted diagnostic.
Event Received: "Mod Diags: 02.01"  Event generated: 18:50:26
Jul 09 1999 Entry: 00002  Slot: 18.02  Id: 01002  Severity:
Inform  Type: Inform Module Diagnostics: Slot 02, Subslot 01.

CB9000>

Event Received: "Diag: PASSED"  Event generated: 18:51:10 19
Jul 09 1999 Entry: 00003  Slot: 02.01  Id: 00103  Severity:
Inform  Type:   Diag Test: 01.00  Loop: 00001

CB9000>

Event Received: "Mod Up: 02.01"  Event generated: 18:52:15 19
Jul 09 1999 Entry: 00005  Slot: 18.02  Id: 01002  Severity:
Inform  Type: Inform Module Up: Slot 02, Subslot 01.
```

Reporting Diagnostic Errors

If the servdiag test encounters an error, and if it is set to stop on the error, the module does not function. If this occurs, call your 3Com reseller or 3Com Technical Support immediately to obtain assistance.

See "The cont_mode Characteristic" later in this chapter for information about how to set the servdiag diagnostic tests to stop on different types of errors.

Setting servdiag Characteristics

With the set servdiag command, you can specify how the EME executes the diagnostics tests on the module that you specify. The characteristics are:

- cont_mode
- loop_count
- verbosity

The cont_mode Characteristic

The cont_mode (continuation mode) of the diagnostic test determines whether the test continues after it encounters an error. The continuation mode can be one of the following:

- continue — The test reports an error then proceeds to the next test after it encounters the error.
- halt_on_fatal — The test stops when it encounters a fatal error. The module is no longer functional. This is the default continuation mode.
- halt_on_nonfatal_and_fatal — The test stops when it encounters a nonfatal error or a fatal error. The module is not longer functional.

The following example sets the cont_mode to continue:

```
CB9000> set servdiag cont_mode continue
```

The loop_count Characteristic

The loop_count characteristic determines how many times the EME runs the diagnostic test on the module that you specify. Because the module does not pass network traffic during the tests, do not set a high loop count when you need to use the module in your network.

Valid values for the loop_count characteristic are 0 through 65535. The default loop_count value is 1. A value of zero causes the test to run indefinitely. The following example sets the loop_count to 5:

```
CB9000> set servdiag loop_count 5
```

The `verbosity` Characteristic

The `verbosity` characteristic determines the amount of output that the diagnostic test sends to the console. Two options are available:

- `normal` — The test reports results at the end of the test or when it encounters an error. This is the default.
- `verbose` — The test reports results at the end of each subtest as well as when it encounters an error and at the end of the test.

The following example sets the `verbosity` to `verbose`:

```
CB9000> set servdiag verbosity verbose
```

Displaying `servdiag` Characteristics

Use the `show servdiag` command to view the characteristics of this option:

```
CB9000> show servdiag
Verbosity:       nonverbose
Loop count:      1
Continue mode:   continue
```

Obtaining Technical Assistance

To receive assistance for installing and troubleshooting the EME, call your 3Com reseller or the 3Com Customer Service Organization. Be prepared to supply a representative with the following information:

- A description of the problem
- The steps that you have taken to try to correct the problem
- The status of the front panel LEDs (if relevant)
- The screen information (if available)
- The configuration of your chassis (the types of modules that are installed by slot)
- The version number of the EME software and switching module software that is operating

5 MANAGING THE CHASSIS POWER AND TEMPERATURE

This chapter describes how to configure and manage the chassis power and temperature parameters using commands from the Management Module. This chapter contains the following topics:

- Managing Power in the Chassis
- Load-Sharing Power Supplies
- Budgeting Power
- Overheat Conditions
- Saved Power Management Configurations
- Displaying Operating Conditions

The management interfaces display "`cb9000`" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.

Managing Power in the Chassis

The chassis provides a fault-tolerant, managed, intelligent power supply subsystem. This subsystem supports:

- Load-sharing power supplies
- High power availability
- EME-based power verification features that are designed to ensure optimal performance

Intelligent Power Subsystem Features

The intelligent power subsystem includes the following features:

- **Load-sharing power supplies** — Provides evenly distributed power consumption among all installed power supplies. Chassis activity disruption is minimized if a power supply fails because there is no changeover (and hence, no changeover interval).
- **Front-loading and rear-loading accessibility** — Provides easy access for upgrades. As your power needs increase over time, it is easy to upgrade by adding a power supply into the rear of the 7-slot chassis. Because power supplies are modular and plug into the backplane, replacing a faulty power supply is a quick procedure.
- **High power capacity** — The power mode and the amount of power available determine the current power limit. The actual power that is delivered depends on whether you are running in non-fault-tolerant mode or in fault-tolerant mode.

  (For detailed information about power supply capacity, see "Power Non-Fault-Tolerant Mode" and "Power Fault-Tolerant Mode" later in this chapter.)

  If a power supply fails while the chassis is running in fault-tolerant mode, still-functioning power supplies provide all of the power necessary to keep installed modules and the chassis running.
- **Software-driven, power management** — The EME polls each switch fabric module and interface module that is installed in the chassis to confirm that enough power is available for new module to power on. If available power is:
  - **Adequate** — The new module powers on.
  - **Inadequate** — The new module does not power on. System overload is avoided.

Software-driven power management also provides protection against the possibility of a catastrophic power failure. If the chassis is operating in power non-fault-tolerant mode and a power supply fails, installed EMEs power off selected (low power class) modules until the power deficit is corrected. Intelligent Power Management ensures that key components and resources continue to operate, even under extreme failure conditions.

(Alerts and traps about changes are sent to the network management applications.)

Load-Sharing Power Supplies

When you determine the total power budget in your system, consider the *system overhead*. System overhead includes power that the chassis itself and its components (fans, backplane signalling, and the EME) consume. Calculate the total power requirements for all installed modules before you install any new module in the chassis. To determine each new module's power requirements, see the documentation that is supplied with each module.

Table 26 lists the system overhead for the chassis.

Table 26 System Overhead in the Chassis

	+5 V	+3.5 V	+12 V	-12 V	-5 V	+2 V
7-slot chassis	29 W	0 W	49 W	0 W	0 W	4 W

V = volts; W= watts

This section describes available power modes and power supply capacity in each power mode. The chassis runs in either of two power supply output modes:

- Power Non-Fault-Tolerant Mode
- Power Fault-Tolerant Mode

Power Non-Fault-Tolerant Mode

Power non-fault-tolerant mode is:

- A user-selectable mode in which 100 percent of the power that can be allocated to modules is available to them (no power is held in reserve).
- The default mode for power supplies as shipped.

While the chassis is running in power non-fault-tolerant mode, the amount of power that is available to modules is determined only by the number of power supplies that are installed. If a power supply fails while the chassis is running in non-fault-tolerant mode:

- Installed modules continue to operate without interruption if the output of the remaining power supplies is sufficient to provide adequate power to all installed modules.
- The EME may shut down selected switch fabric modules and interface modules to bring installed module power consumption under the now-reduced power budget. See "Budgeting Power" on page 118 for more information about how the EME manages power.

Power Fault-Tolerant Mode

Power fault-tolerant mode is a user-selectable mode in which power that is equivalent to one power supply is held in reserve. This reserve power is not available to installed modules unless a power supply fails, or if you switch the power mode from power fault-tolerant mode to power non-fault-tolerant mode.

While the chassis is running in power fault-tolerant mode:

- All installed power supplies are functioning and contributing power to the chassis and modules. No single power supply is a dedicated standby power supply. Rather, a factory-defined power limit ensures that power that is equivalent to at least one power supply is available to replace power lost if a power supply fails.
- The amount of power that installed modules require must not be greater than the number of installed power supplies, minus one (n-1). When you reserve power that is equivalent to one power supply in power fault-tolerant mode, the failure of a single power supply has no impact on installed modules that are already powered on.

If a power supply fails while the chassis is running in fault-tolerant mode:

- The EME automatically disables fault-tolerant mode.
- Power formerly reserved is made available by power class and slot location to power-enabled modules to prevent them from powering off (as an attempt to bring power consumption under the now-reduced power budget).
- All modules that had power before the power supply failure continue to receive power without interruption.
- Upon power supply recovery (or replacement), the EME automatically reenables fault-tolerant mode.

Setting Power Fault-Tolerance

By default, the chassis is set to non-fault-tolerant mode. To set the chassis to power fault-tolerant mode or to power non-fault-tolerant mode, enter `set power mode` at the EME prompt.

Use the following syntax:

```
set power mode fault_tolerant
set power mode non_fault_tolerant
```

The following example sets the power mode to fault-tolerant:

```
CB9000> set power mode fault_tolerant
Power will switch to FAULT_TOLERANT mode when sufficient
power is available
```

When you attempt to set the chassis to power fault-tolerant mode, the EME determines if sufficient unallocated power exists to place one power supply's worth of power in reserve:

- If the unallocated power budget is sufficient, the chassis sets to power fault-tolerant mode.
- If the unallocated power budget is not sufficient, the chassis remains in power non-fault-tolerant mode.

Enabling and Disabling Power to Slots

You can enable or disable power to any slot in your chassis, and the EME does not turn on power to the module in the disabled slot. Modules in disabled slots are not allocated power. All slots are enabled by default.

You enable or disable power to slots by entering the `set power mode` command at the EME prompt using the following syntax:

```
set power slot <slot #> mode enable
set power slot <slot #> mode disable
```

In the following example, power is enabled to slot 2:

```
CB9000> set power slot 2 mode enable
Slot 2 enabled
CB9000>
```

If there is:

- *Sufficient power* available to meet the requirements of the new module, the EME enables power to the specified slot and reduces the power budget by the amount of power that module consumes.
- *Insufficient power* to meet the requirements of the new module, the module remains in power-pending state until sufficient power becomes available.

A module that was powered off due to a lack of sufficient available power is in *power pending state*. The EME automatically powers on the module again when sufficient power becomes available.

Power Class Settings

A *power class setting* is a value in the range of 1 through 10 that is assigned to each module. The highest setting is 10. Each module has a default power class setting, which you can change with an EME command. The EME uses the power class settings to manage power among the modules in the chassis, and to determine the order in which it powers on and powers off installed modules.

Even though the EME has a power class setting, you cannot manage the power of an EME module. An EME always draws power when it is inserted in the chassis, and you cannot power off an EME module using an EME command.

Using the Default Power Class Setting

Each module is shipped with a default power class setting:

Module	Default Power Class Setting
EME	10
EMC	10
Interface Module	3
Switch Fabric Module	9

Setting Power Class

To set the power class for a module that is in a specified slot, enter the `set power class` command at the EME prompt using the following syntax:

```
set power slot <slot #> class <class #>
```

In the following example, the module in slot 2 is set to power class 5:

```
CB9000> set power slot 2 class
Enter class: 5

slot 02 power class is set to 05.
CB9000>
```

Power Class 10 Warnings

The EME cannot automatically power off a module that is assigned a power class setting of 10.

For example, if a power supply failure causes a power deficit (or if a chassis overheat condition develops), a module that is assigned a power class setting of 10 continues to run until you order it to power off. Under some conditions (such as an extended overheat condition), chassis or module hardware damage may result.

CAUTION: *To ensure that the EME can make all power management decisions automatically, do not assign a power class setting of 10 to any switch fabric module or interface module unless it is absolutely necessary.*

Budgeting Power

This section describes:

- Allocating Power for Installed Modules
- Increasing the Unallocated Power Budget
- Determining Chassis Power Budget
- Power Supply Output in Non-Fault-Tolerant Mode
- Power Supply Output in Fault-Tolerant Mode

Allocating Power for Installed Modules

Before you install a new module in the chassis, use the `show power budget` command to confirm that there is sufficient power for installed modules. The `show power budget` command displays current chassis power conditions that help you decide if there is sufficient power available to power on and operate the new module.

Table 27 shows selected EME power management commands and their functions.

Table 27 Selected EME Power Management Commands

Command Name	Displays
`show power budget`	Power budget information on a per-voltage basis. Displays actual voltages measured on the backplane.
`show power slot <slot #>` or `<all>`	Power class, power state, and power status for the module in a specified slot, or in all slots.

Table 27 Selected EME Power Management Commands

Command Name	Displays
`show power mode`	Whether power fault-tolerant or power non-fault-tolerant mode is currently in effect for the chassis. Also indicates whether `overheat_auto_power_down` is enabled or disabled.
`show power all`	All information that the preceding commands display.

The EME provides initial module power consumption values from the power consumption table that it maintains:

- When an EME powers on a module, it adjusts the available power budget to reflect the power consumption of the newly powered-on module.
- The EME then powers on remaining modules (by power class and slot location) to the limit of the unallocated power budget.

By maintaining an accurate power budget, an EME can determine which installed modules to:

- Power on.
- Power off to bring module power consumption under budget (if any).
- Place in power pending state due to a lack of sufficient unallocated power budget to power them on.

Increasing the Unallocated Power Budget

This section describes actions that you can take to increase the unallocated power budget whenever you need more power for installed switch fabric modules and interface modules, or to power on newly installed modules.

To increase the unallocated power budget:

1. Add one or more power supplies.

   For instructions and information, see the *7-Slot Chassis Power Supply Installation Guide*, which is available on the Switch 4007 Software and Online Manuals CD or from the 3Com Web site.

2. If the chassis is running in power fault-tolerant mode, change the power mode to power non-fault-tolerant to make reserve power available to all installed modules.
3. Manually power off selected modules until you have enough power.

Determining Chassis Power Budget

To ensure optimal power fault-tolerance, determine the current power budget for the chassis as follows:

1 At the terminal prompt, enter: **`show power budget`**

The `show power budget` command shows the amount of power currently available for modules:

- Total power installed
- Amount of power consumed
- Amount of power available

2 Examine the output of the `show power budget` command. If necessary, add another power supply to your chassis to provide sufficient additional power to enable power fault-tolerant mode.

Example:

```
CB9000> show power budget

         Power Management Information
         ----------------------------

Chassis Power Budget :

Voltage Type  Voltage Level  Watts Capacity   Watts Available   Watts Consumed
------------  -------------  --------------   ---------------   --------------
    +3V          3.556         1154.00           517.00           637.00

    +5V          5.281         1184.00           565.00           619.00

 +3V+5V Shared     N/A         1310.00           568.00           742.00

    -5V         -5.001           30.00            14.50            15.50

   +12V         12.066          240.00            30.50           209.50

   -12V        -12.010           36.00            17.00            19.00

    +2V          2.154           16.00             4.00            12.00
```

Power Supply Output in Non-Fault-Tolerant Mode

In Table 28, values are rounded values that do not include system overhead (fans, backplane, signalling, and EMEs). Table 28 shows the power available in power non-fault-tolerant mode (by voltage type) when the power supply is 930 watts.

Table 28 Power Output in Non-Fault-Tolerant Mode (7-slot Chassis)

Output Voltage (Volts)	1 Power Supply (Watts)	2 Power Supplies (Watts)
+3	682	1364
+5	210	449
+3 and +5	821	1671
+12	22	94
+2	4	12
TOTAL WATTS	1739	3590

Power Supply Output in Fault-Tolerant Mode

In Table 29, values are rounded values that do *not* include system overhead (fans, backplane signalling, and EMEs). Table 29 shows the power available in power fault-tolerant mode (by voltage type) when the power supplies are 820 watts.

Table 29 Power Output in Fault-Tolerant Mode (7-slot Chassis)

Output Voltage (Volts)	1 Power Supply (Watts)[1]	2 Power Supplies (Watts)
+3	N/A	577
+5	N/A	567
+3, +5	N/A	630
-5	N/A	14.5
+12	N/A	46
-12	N/A	17
+2	N/A	4
TOTAL WATTS	N/A	1855.5

[1] Power fault-tolerance can only be established if at least one power supply's worth of unallocated power budget is available to be held in reserve.

Overheat Conditions

An overheat condition exists when one of the chassis temperature sensors detects a chassis internal operating temperature that exceeds a predefined threshold. The allowable ambient temperature operating range is 0 °C through 50 °C (32 °F through 122 °F). The default threshold setting is fixed at an upper limit of 60 °C (140 °F) or higher to prevent module damage.

Cooling loss or excessively high ambient (room) air temperature can cause an overheat condition.

The following events occur during an overheat condition:

1. The Primary EME character display shows the word `TEMP`
2. If an SNMP agent is present in the chassis, power management informs the SNMP agent of the overheat condition.
3. A 1-minute delay is provided, during which the Primary EME and external management entities are notified of the overheat condition.
4. Approximately 1 minute later, the EME initiates a power-off strategy to all modules installed in the overheat management areas where the overheat condition was detected.
5. The overheat indication `TEMP` stops when the chassis internal operating temperature falls below the temperature threshold and stays there for 15 minutes.

The EME does *not* power off modules that occupy slots outside of affected overheat management areas. This overheat power-off strategy is based on the power class setting and slot location of each installed switching module.

Enabling and Disabling Automatic Module Power-off

To enable automatic module power-off in response to an overheat condition, use the `set power overheat_auto_power_down` command as follows:

```
set power overheat_auto_power_down mode enable
set power overheat_auto_power_down mode disable
```

The two overheat auto-power-down modes are:

- **Enable** — Causes slots to power off automatically when the chassis overheats.
- **Disable** — (default) Causes the EME to send notification to network management applications, but the chassis keeps operating.

CAUTION: *If* `set power overheat_auto_power_down mode disable` *is in effect when an overheat condition occurs, the chassis and all installed, powered-on, modules continue to run. Under these circumstances, an extended overheat condition may cause heat-related hardware damage. 3Com recommends that you run the chassis with* `overheat_auto_power_down enable` *in effect.*

In the following example, overheat power-down mode is enabled:

```
CB9000> set power overheat_auto_power_down mode enable

overheat-power-down mode set to ENABLE
CB9000>
```

The Overheat Management Area

The overheat power-off process in a 7-slot chassis is based on one temperature sensor that treats the module payload area of the chassis as one overheat management area. Modules power off in this overheat management area according to their power class settings.

Overheat Power-off Process

The module overheat power-off process is as follows:

1. When any chassis temperature sensor detects an internal chassis operating temperature of 45 °C (113 °F) or higher, power management issues warning traps that inform the user that an overheat condition may soon exist. The system generates warning traps every 30 seconds (approximately) at this point.
2. When internal chassis operating temperature reaches 60 °C (140 °F), power management power-disables *selected* modules installed within the affected overheat management area to reduce the 5-volt power consumption by at least 50 watts.

   Selected modules in the overheat management area power off, in order, starting with modules that have the lowest power class setting.

   This reduction of power consumption should provide a 2 °C drop in temperature per slot at the temperature sensor for the overheat management area. The system generates overheat traps every 10 seconds (approximately).
3. If two or more modules in the affected overheat management area have the *same* power class, they power off from highest slot number to lowest slot number.
4. Switch fabric modules and interface modules continue to power off until all modules in the affected overheat management area have powered off or until you resolve the overheating condition. Modules with a power class setting of 10 continue to run.
5. Chassis temperature is allowed to stabilize for 15 minutes before further action is taken.
6. If chassis temperature is not at or below the established overheat threshold after 15 minutes have elapsed, *all* modules are powered off. Modules do not power on again until you correct the overheat condition.

Overheat Recovery Process

Overheat recovery occurs when the temperature sensor that detected an overheat condition reports that internal chassis temperature is at or below the overheat threshold.

When overheat recovery is initiated, modules that were powered off to alleviate the overheat condition power on to the limit of the current power budget.

The overheat recovery process proceeds as follows:

- The EME powers on modules, in order, from the lowest slot number to the highest slot number.
- The EME powers on modules with the highest power class setting first. If two or more modules have the *same* power class setting, they power on from the lowest slot number to the highest slot number.

Saved Power Management Configurations

The EME stores:

- Saved power management configuration data for all installed network modules in on-board EME non-volatile RAM (NVRAM).
- Unmanaged power allocation data that describes the type (per voltage) and the amount of power (watts) that are available to installed modules.

When the chassis powers on or after a chassis reset:

- The EME uses saved power management configuration data to verify that power configurations for installed modules precisely match those in effect prior to the chassis reset.
- If necessary, the EME uses the saved data to restore lost module power configurations.

The EME saves the power management configuration data that is listed in Table 30.

Table 30 Saved Power Management Configuration Data

Data Type	Descriptions
Slot profile	Identifies the module installed in a given slot. In addition, empty slots are identified.
Slot power state	Power state for each installed module (enabled, disabled, or pending).
Slot power class	Power class setting for each installed module.
Power mode	Power mode for the chassis prior to a chassis reset (power fault-tolerant mode or power non-fault-tolerant mode).
Overheat auto-power-down mode	Auto-power-down mode of the chassis prior to a chassis reset (enabled or disabled).

When the chassis powers on or after a chassis reset, the EME compares saved slot profile data for the modules that are installed in each successive slot with current slot profile data for those same modules. Module power is based on power class setting and relative slot location.

CAUTION: *If you power off the chassis, then move or install modules while the chassis is powered off, the system reevaluates chassis power management based on available power in the chassis.*

Displaying Operating Conditions

Use variations of the `show` command to display chassis, module, and port operating conditions and to identify installed chassis components.

Displaying Chassis Information

Use the `show chassis` command to display basic information about chassis operating conditions, including temperature and power supply conditions. The following information is provided by this command:

- **Type** — The specific model of a chassis
- **Backplane** — The type and revision level of the backplane
- **Power supply** — If a power supply is present in a slot, its normal or faulty status, and its model number
- **Fan** — The status of each chassis fan tray
- **Temperature** — Chassis temperature at three locations

Displaying Module Information

Use the `show module` commands to display status information for a module and submodule that is installed in a specific slot or to display information for *all* modules and submodules that are installed in the chassis.

The following `show module` commands are available:

- `show module <slot.subslot>`
- `show module all`
- `show module all verbose`

The subslot number is always 1.

Basic Information For One Module

To display basic information for a module installed in a specific slot, use the following command syntax:

```
show module <slot.subslot>
```

Where `<slot.subslot>` is the location of the module in the chassis. The following example displays basic information for the EME installed in slot 17, subslot 1 in a 16-slot chassis:

```
CB9000> show module 17.1

Slot    Module     Status  Description
-----   ------     ------  -------------------------------
17.01   3CB9EME    Active  Enterprise Management Controller
```

Basic Information For All Modules

To display basic information for all installed modules, use the `show module all` command.

Detailed Information For All Modules

To display detailed information about all of the modules that are installed in your chassis, including information about module software and DIP switch settings, use the `show module all verbose` command.

Displaying Power Information

Use the `show power` commands to display the power budget, power modes, and power information on a per-slot basis.

Table 31 lists the commands that display current power conditions in the chassis.

Table 31 Commands Used to Display Current Power Conditions

Command	Description
`show power budget`	Indicates how power output is distributed among all installed load-sharing power supplies. This information helps you to determine if chassis power is sufficient to permit the addition of modules, and to avoid an unintentional loss of power fault-tolerance (if currently in effect).
`show power mode`	Indicates which of two power modes is currently in effect (fault-tolerant or non-fault-tolerant).
`show power slot`	Displays the slot number, power class setting, administrative status (slot power enabled or disabled), and module operating status of a module that is installed in a specified slot.
`show power all`	Displays power mode, slot power information, and power budget information for all installed modules.

When you enter the `show power mode` command while the chassis is running in non-fault-tolerant mode, the following information appears:

```
Fault-Tolerant      Mode:      NON_FAULT_TOLERANT
Overheat Power Down Mode: DISABLE
Fault-Tolerant Status:    NON_FAULT_TOLERANT
```

When you enter the `show power mode` command while the chassis is running in fault-tolerant mode, the following information appears:

```
Fault-Tolerant      Mode:      FAULT_TOLERANT
Fault-Tolerant Status:    FAULT_TOLERANT
Overheat Power Down Mode: DISABLE
```

Displaying Chassis Inventory Information

The `show inventory` command displays contents of a chassis, including hardware release numbers and serial numbers. You can display inventory with the following options:

- `show inventory chassis`
- `show inventory module`
- `show inventory power supply`
- `show inventory summary`

Displaying EME Information

The `show eme` command displays various aspects of information that have been configured on the EME. Example:

CB9000> **show eme**

```
Name:
  CoreBuilder-9000
Location:
  Boston
For assistance contact:John Smith
  System Administrator

Operational Version: v3.0              Boot Version:        v3.0

Serial Number:       9ABJ001292        Service Date:        1999/04/
Mac Address:         08-00-8f-30-c7-27    Restarts:         7
Cpu Ram Size (MB):   20                Flash Memory (MB):   16

Trap Receive:        DISABLED          Diagnostics:         ENABLED
```

III

UNDERSTANDING YOUR SWITCHING MODULES

6 MODULE PARAMETERS

This chapter provides guidelines and other key information about how to implement module parameters.

The chapter covers these topics:

- Module Parameters Overview
- Key Concepts
- If the module being hot-swapped is the same type as the module that was removed from a chassis, the hot-swapped module learns the previous module's settings from the EME module that is installed in the chassis.
- nvData

You can manage module parameters in these ways:

- *From the* `module` *menu of the Administration Console. (See the* Switch 4007 Command Reference Guide.*) You can use the Administration Console after you log in to the EME and connect to a module slot.*
- *From the Web Management software. (See the* Switch 4007 Getting Started Guide.*)*

The management interfaces sometimes display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Module Parameters Overview

You use the module parameters to set values for specific functions, or to modify values that are set on a module during power-on. You can modify only some of the module parameters values.

Features

Using the module parameters, you can set or modify the factory default values for a module in your Switch 4007 system for the following functions:

- Display the module's current configuration
- Take a snapshot of a module's current configuration and status
- Create and administer a statistics baseline

  See the *Switch 4007 Command Reference Guide* for detailed information about creating and maintaining baselines.

If you set a baseline while connected to a module and then disconnect from that module, the baseline is disabled. You must reconnect to the module and use the `requestedState` *option to change the baseline value for that module to* `enable`*.*

- Administer nvData

This chapter discusses the redundancy and administering nvData options.

You can also use the module parameters to:

- Modify the module name
- View the module's date and time
- Clear the module's diagnostics block
- Reboot the module

See the *Switch 4007 Command Reference Guide* or online Help for more information about how to set and modify module parameters.

Benefits

Using the module parameters to set and modify module functions:

- Provides an easy method for both setting and modifying module-level functions
- Decreases the time and cost that it takes to modify a module configuration if you have to constantly make changes from the same source then reboot the module to apply the changes

Key Concepts

This section explains how to set and modify module values for applicable parameters and defines terms that are used during each process.

How to Set and Modify Module Parameters

The module parameter values are set at the factory or during power-on. The basic steps for setting or modifying module parameter values are:

1. Connect to a module. The `menu options` screen appears.
2. Select `module` from the `menu options` screen. The module `menu options` screen appears.
3. Select an option from the module `menu options` screen. The screen for that option appears with information and other options, if applicable.
4. Read the screen for details explaining how to modify or quit the module `menu options` screen.

Terminology

Before you use the module parameters to set or modify function values, review the following terms:

- **Reset** — Option in the `nvData` menu. Use this option to reset module values to their factory defaults.
- **Staging** — Option in the `nvData` menu. If disabled, the module resets to its default factory settings after you hot-swap it into another Switch 4007 chassis. If enabled, the module retains its nvData settings after you hot-swap it into another Switch 4007 chassis.

If the module being hot-swapped is the same type as the module that was removed from a chassis, the hot-swapped module learns the previous module's settings from the EME module that is installed in the chassis.

nvData

When you work with nvData, you can:

- Restore the module's nvData values to their factory defaults, using the `reset` option. Use the `reset` option if you have problems with a new setting and want to start over again.
- Prepare the system (update the image name and IP address) for an emergency download using the `emergencyDownload` option.
- Display the last download performed on the module you selected using the `displayDownload` option.
- Save the nvData settings when you hot-swap a module, using the `staging` option. The nvData settings are saved when you set the `staging` bit on a module to `enable` and hot-swap that module to another Switch 4007 chassis.

If you hot-swap a module with the `staging` *option set to* `disable` *to a Switch 4007 chassis slot where the same type of module was previously installed, the module learns the previous module's settings from the EME module. Otherwise, the module is reset to its factory default settings.*

7 Physical Port Numbering

This chapter provides guidelines and other key information about port numbering in the Switch 4007 system.

The chapter covers these topics:

- Slot Architecture
- Default Port Settings
- Allocating Switch Fabric Capacity to Slots
- Key Guidelines for Implementation
- Effects of Removing a Module
- Effects of Replacing Modules

Slot Architecture

The slots in the Switch 4007 chassis are designed to hold the following components:

- Slots 8 and 9 are reserved for the primary and secondary Management Modules (Enterprise Management Engines, or EMEs).
- Slot 7 is reserved for the switch fabric module.

  The backplane in the Switch 4007 chassis uses a star-wired interconnect configuration to connect the switch fabric module slot to all other switching module slots.
- Slots 1 – 6 are reserved for Layer 2 and Multilayer Switching Modules.

  You can install any supported switching module in any of these slots, however one consideration is the capacity of the switch fabric module in Slot 7. For example, if you want to install multiple 4-port Gigabit Ethernet Multilayer Switching Modules, consider installing the 24-port switch fabric module instead of the 9-port switch fabric module.

Default Port Settings

Table 32 describes the default backplane port settings of Switch 4007 modules.

Table 32 Default Port Settings

Module	Factory Default Setting for Front Panel Ports	Factory Default Setting for Backplane Ports	Notes
Switch Fabric Modules			
3CB9FG24T — 24-Port Gigabit Ethernet Switch Fabric Module	—	■ backplane ports 1 – 24: enabled (all slots)	See Table XX for how ports are connected to other slots in the chassis.
3CB9FG9 — 9-Port Gigabit Ethernet Switch Fabric Module	Ports 7 – 9 are enabled when GBICs are installed.	■ backplane ports 1 – 6: enabled (all slots)	See Table XX for how ports are connected to other slots in the chassis.
Layer 2 Switching Modules			
3CB9LF36R — 36-Port 10/100BASE-TX Fast Ethernet RJ-45 Layer 2 Switching Module	enabled	■ Port 37: enabled ■ Port 38: disabled	With a 24-port switch fabric, both backplane ports on this module can carry traffic. WIth a 9-port switch fabric, only one backplane port can carry traffic.
3CB9LF20MM — 20-Port 100BASE-FX (MT-RJ) Fast Ethernet Layer 2 Switching Module	enabled	■ Port 21: enabled ■ Port 22: disabled	With a 24-port switch fabric, both backplane ports on this module can carry traffic. With a 9-port switch fabric, only one backplane port can carry traffic.
3CB9LG9MC — 9-port Gigabit Ethernet Layer 2 Switching Module	enabled	■ Port 10: enabled ■ Port 11: disabled ■ Port 12: disabled	With a 24-port switch fabric, all backplane ports on this module can carry traffic. With a 9-port switch fabric, only one backplane port can carry traffic.
Multilayer Switching Modules			
3CB9RG4 — 4-port Gigabit Ethernet Multilayer Switching Module	enabled	■ Port 5: enabled ■ Port 6: disabled ■ Port 7: disabled ■ Port 8: disabled	With a 24-port switch fabric, all backplane ports on this module can carry traffic. With a 9-port switch fabric, only one backplane port can carry traffic.
3CB9RF12R — 12-Port 10/100BASE-TX Multilayer Switching Module	enabled	■ Port 13: enabled	With either switch fabric module, the module has one port to carry traffic.

Table 32 Default Port Settings (continued)

Module	Factory Default Setting for Front Panel Ports	Factory Default Setting for Backplane Ports	Notes
Interface Modules (no on-board switching)			
3CB9LG4 4-Port GBIC Gigabit Ethernet (GEN) Interface Module	Front panel ports are enabled when GBICs are installed.	Uses the switch fabric module backplane ports.	These modules do not switch traffic, but rather act as a traffic pipeline, passing traffic from their front panel ports directly to the backplane ports on the switch fabric module.

Configuring Port Status

You can enable (place on-line) or disable (place off-line) Switch 4007 ports. When a port is enabled, frames are transmitted normally over that port. When a port is disabled, the port neither sends nor receives frames.

CAUTION: *Before you can configure two backplane ports as trunk ports, both backplane ports must be enabled. Enabling both backplane ports, (except for the 1000BASE-SX and 1000BASE-LX interface modules) can cause a network loop. Enable both backplane ports only if you plan to configure the two ports as a trunk port. See Table 32 for information about backplane port settings, and Chapter 12 for information about configuring trunk ports.*

Allocating Switch Fabric Capacity to Slots

This section describes how the capacities of the 9-port and 24-port Gigabit Ethernet Switch Fabric Modules are distributed to the other switching module slots in the chassis.

9-port GEN Switch Fabric Module

The capacity of the 9-Port Gigabit Ethernet (GEN) Switch Fabric Module (3CB9FG9) is distributed across the slots in the chassis. The module capacity is expressed in terms of *backplane ports*.

When a switching module is installed, its rear ports communicate with the switch fabric module backplane ports that are assigned to its slot.

The module's six backplane ports are assigned to the chassis slots as indicated in Table 33.

Table 33 Mapping the 9-Port GEN Switch Fabric Module (SFM)

Chassis Slot No.	SFM backplane port number that is assigned to the slot	SFM LED that is assigned to the backplane port
1	1	1
2	2	2
3	3	3
4	4	4
5	5	5
6	6	6
7	Not applicable. Switch fabric module slot only.	

Using Table 33: Examples

Suppose you installed a 9-port Switch Fabric Module. Next, you installed a 36-port Fast Ethernet Layer 2 Switching Module in slot 4. After both module's boot sequences were finished, you should see a LED#4 on the switch fabric module to be lighted green. This indicates that the switching module recognizes that there is a module in slot 4.

Next, in that same chassis, suppose you installed a 4-port Gigabit Ethernet Multilayer Switching Module in slot 5. After it completed its boot sequence, you would look for LED#5 to become lighted on the switch fabric module. Notice that, although the module in slot 5 has four backplane ports (see Table 32), only one of them can pass traffic to the switch fabric module.

24-port GEN Switch Fabric Module

The capacity of the 24-Port Gigabit Ethernet Switching Fabric Module (3CB9FG24T) is distributed across the slots in the chassis. The module capacity is expressed in terms of *backplane ports*. When a switching module is installed, its rear ports communicate with the switch fabric module backplane ports that are assigned to its slot. The module's 24 backplane ports are assigned to the chassis slots as indicated in Table 34.

Table 34 Mapping the 24-port Switch Fabric Module (SFM)

Chassis Slot No.	SFM backplane port numbers that are assigned to the slot	SFM LEDs that are assigned to the backplane ports
1	1	1
	2	2
	3	13
	4	14
2	5	3
	6	4
	7	15
	8	16
3	9	5
	10	6
	11	17
	12	18
4	13	7
	14	8
	15	19
	16	20
5	17	9
	18	10
	19	21
	20	22
6	21	11
	22	12
	23	23
	24	24
7	Not applicable. Switch fabric module slot only.	

Using Table 34: Examples

Suppose you installed a 24-port Switch Fabric Module. Next, you installed a 36-port Fast Ethernet Layer 2 Switching Module in slot 4. After both module's boot sequences were finished, you should see LED #7 and #8 on the switch fabric module to be lighted green. This indicates that the switching module recognizes that there is a module in slot 4. However, from Table 34, notice that the corresponding switch fabric port numbers are 13 and 14, respectively. Also notice that port numbers 15 and 16 on the switch fabric module (and LEDs 19 and 20) will not be used.

Next, in that same chassis, suppose you installed a 4-port Gigabit Ethernet Multilayer Switching Module in slot 5. After it completed its boot sequence, you would look for LED# 9, 10, 21, and 22 to become lighted on the switch fabric module. However, from Table 34, notice that the corresponding switch fabric port numbers are 17, 18, 19, and 20, respectively.

Key Guidelines for Implementation

To ensure that you understand the port numbering that the system reports for certain aspects of your configuration (bridging information, trunks, and virtual LANs), observe these guidelines when you configure your system:

- Before you attempt to configure any bridging parameters, determine your physical port configuration.
- If you use *trunking* to group ports, configure your trunks *before* you attempt to configure any Virtual LANs (VLANs). Be sure that you understand how trunking associates a group of ports with a trunk. (See Chapter 12.) These associations affect the following situations:
  - When you perform an operation for which you must specify bridge ports (for example, when you define VLANs), you must use the lowest-numbered port in each trunk to represent the trunk. The operation that you perform then applies to all ports in the trunk.
  - When you view information that applies to more than one port (for example, bridging displays for trunks), the port number field identifies all ports in the trunk. A VLAN summary display lists all physical ports to indicate which physical system connectors can receive or transmit frames within a VLAN. (Use the VLAN detail display to see trunk port groups.)

Effects of Removing a Module

When you remove a module and leave the slot empty, a number of changes occur.

VLAN Changes

When you remove a module, VLAN changes occur as follows:

- If you have a VLAN that includes ports that are associated with the removed module, those ports are not actually removed from the VLAN, but transition to the link down state.

  Example:

  If a VLAN contains ports 1 through 3 in a module in slot 3, removing the module places ports 1 through 3 in the VLAN in the link down state.
- If there are no remaining ports in the VLAN after you remove the module, the VLAN summary display lists all ports for the VLAN as being in the link down state.

See Chapter 14 for more information about VLANs.

Trunk Changes

When you remove a module, trunk changes occur as follows:

- There are no trunking changes to the switch fabric module when you remove a module. The ports remain in the trunk, but the trunk is down.
- You cannot access the trunk summary display on the module because it is no longer there.

See Chapter 12 for more information on trunking.

Effects of Replacing Modules

When you remove a module and replace it with another module, a number of changes can occur, depending on the replacement module.

Replacing Modules of the Same Type

If you remove a module and replace it with a module of the same type (model number), the following occurs:

- Port numbering is not affected — you can exchange modules without affecting the port numbers.
- The system remembers ports that were members of trunks and VLAN. When you insert another module into the slot, the ports are added back into the trunks and VLANs.

Example:

If you have a 20-port Fast Ethernet module with ports 1 and 2 assigned to a VLAN, and then replace it with a new 20-port Fast Ethernet module, the new module takes on the configuration of the previous module.

Replacing Modules of Different Types

More complicated changes occur when you replace modules of different types (different model numbers).

Swapping a 20-port Fast Ethernet Layer 2 Switching Module with a 12-port Fast Ethernet Multilayer Switching module has the following effects:

- Port numbering is reduced to 10 ports.
- The new module configures to its default VLAN settings. The system does not remember which ports were part of a previous VLAN. You must redefine any new VLANs.
- The ports from any previous trunk configurations are dropped. You must redefine any new trunks.

  Example:

  If you replace a Fast Ethernet Switching Module in slot 1 (that has a trunk on ports 5 and 6) with an Fast Ethernet Multilayer Switching Module, the new FX ports 5 and 6 do *not* become part of the trunk. For more information about trunking, see Chapter 12. For information about VLANs, see Chapter 14.

8 ETHERNET

This chapter provides guidelines and other key information about how to implement Ethernet ports.

The chapter covers these topics:

- Ethernet Overview
- Key Concepts
- Key Guidelines for Implementation
- Port Enable and Disable (Port State)
- Port Labels
- Autonegotiation
- Port Mode
- Flow Control
- PACE Interactive Access
- Port Monitoring
- Standards, Protocols, and Related Reading

You can manage Ethernet features in either of these ways:

- *From the* `ethernet` *menu of the Administration Console. (See the* Switch 4007 Command Reference Guide*.) You can use the Administration Console after you log in to the Enterprise Management Engine and connect to a module slot.*
- *From the Ethernet folder of the Web Management software. (See the* Switch 4007 Getting Started Guide*.)*

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Ethernet Overview

Ethernet is a standardized, switched, packet-based network that supports an exponential hierarchy of three line speeds:

- **10 Mbps** — Ethernet
- **100 Mbps** — Fast Ethernet
- **1000 Mbps** — Gigabit Ethernet

All speeds of Ethernet are based on an IEEE 802.3 standard protocol called Carrier Sense Multiple Access with Collision Detection (CSMA/CD), which controls network access. With CSMA/CD, a station that intends to transmit listens for other Ethernet traffic on the network. When the station does not detect network activity, the station transmits.

Features

You can configure these features on Ethernet ports:

- **Port state** — Whether a port is `enabled` (placed online) or `disabled` (placed off-line)
- **Port label** — An alphanumeric port identifier
- **Port mode** — Port speed (10 Mbps, 100 Mbps, or 1000 Mbps) and duplex mode (half-duplex or full-duplex)
- **Autonegotiation** — A feature that allows some ports to automatically identify and negotiate speed and duplex mode with a receiving device
- **Flow control** — A Fast Ethernet or Gigabit Ethernet port mode that pauses and resumes transmissions
- **PACE® Interactive Access** — An algorithm that reduces network jitter, provides reliable timing, and optimizes LAN bandwidth use

In addition, some important Ethernet features depend on which Ethernet equipment you use, how you configure it, and how you connect it:

- **Trunking** — Increases bandwidth between switches and servers
- **Trunk Control Message Protocol (TCMP)** — Increases the availability of trunked links by handling physical configuration errors
- **Gigabit Interface Converter (GBIC)** — A Gigabit Ethernet port media type that allows you to hot-swap one media connector without affecting the other connectors

Benefits

Ethernet, Fast Ethernet, and Gigabit Ethernet technologies allow you to configure and optimize:

- Link bandwidths
- Link availability

Link Bandwidths

As your network needs to support more users and increasingly bandwidth-intensive applications, you can configure Ethernet networks to keep pace with (or exceed) the capacity demands at two locations:

- **To end stations** — Depending on your application needs and network growth, you can migrate workstation connections from shared 10-Mbps to switched 100-Mbps Fast Ethernet. 3Com's Ethernet network interface cards (NICs) can automatically sense and configure themselves to an upgraded connection speed.
- **Between servers and switches** — Ethernet systems allow you to increase the bandwidth between switches or between servers and switches as your network requires. This increase is accomplished using *trunking* technology (also called *link aggregation*), which works at Open Systems Interconnection (OSI) Layer 2. For more information about trunking, see Chapter 12.

Link Availability

Ethernet technologies also allow you to design high levels of availability into your network through the use of trunking. A trunk enhances network availability because its underlying TCMP technology detects and handles physical configuration errors in point-to-point configurations. For more information about trunking, see Chapter 12.

Other Benefits

The hierarchy of Ethernet, Fast Ethernet, and Gigabit Ethernet technologies offers these additional network benefits:

- Ease of configuration and expansion of point-to-point links
- Increased support for workstation moves, adds, changes, and upgrades
- Low-cost expansion of switch-to-switch or switch-to-server bandwidths without having to change device modules or cabling
- With PACE Interactive Access, reduction of network jitter, improved network timing, and optimization of LAN bandwidth use

Key Concepts

These concepts are important to implementing Ethernet:

- **Carrier Sense Multiple Access with Collision Detection (CSMA/CD)** — The standardized Ethernet protocol that controls device access to the network
- **Collision** — When two or more stations attempt to transmit simultaneously
- **Port mode** — An Ethernet port's speed and duplex mode
- **Port speed** — 10 Mbps (Ethernet), 100 Mbps (Fast Ethernet), 1000 Mbps (Gigabit Ethernet)
- **Port state** — Whether a port is enabled (placed online) or disabled (placed off-line)
- **Duplex mode** — Whether a port supports one-way (half-duplex) or two-way (full-duplex) transmissions
- **Autonegotiation** — A feature that allows some ports to identify and negotiate speed and duplex mode with a receiving device.
- **Flow control** — A Fast Ethernet or Gigabit Ethernet port mode that pauses and resumes transmissions
- **Trunking** — A technology that combines multiple Fast Ethernet or Gigabit Ethernet ports into a single high-speed channel, thereby increasing bandwidth between switches and between servers and switches
- **Trunk Control Message Protocol (TCMP)** — A protocol that detects and handles physical configuration errors in a point-to-point configuration, thereby increasing availability of trunked links
- **Gigabit Interface Converter (GBIC)** — A Gigabit Ethernet port media type that allows you to hot-swap one media connector without affecting the other connectors

- **PACE Interactive Access** — An algorithm that controls traffic flow on a point-to-point link with an end station. In a typical half-duplex Ethernet connection, you can never achieve high rates of utilization because of the randomness of collisions. If a switch and end station both try to send data, a collision occurs, forces retransmission, and lowers link utilization.

  PACE Interactive Access enables higher link utilization by altering the switch's *back-off* behavior. Instead of continuing to send data after winning a collision, the switch waits, allows the end station to send a packet, and then retransmits. The result is an interleaving of transmissions between the end station and the switch.

  This feature avoids repetitive collisions and prevents an end station from "capturing" the link. (With conventional Ethernet, a packet collision can cause the last station that transmitted successfully to monopolize Ethernet access and cause delays.)

- **Network areas** — 3Com uses a three-tiered framework to describe the different functional areas in a LAN:

  - **Wiring closet** — This area provides connections to user workstations. It also includes downlinks into the data center or campus interconnect.

  - **Data center** — This area receives connections from wiring closets and campus interconnect areas. Most local server farms reside here.

  - **Campus interconnect** — This area appears as a separate location only in larger networks; smaller networks usually have only wiring closets and data centers. The campus interconnect links campus data centers to each other. It may also include an enterprise server farm and connections to a wide area network.

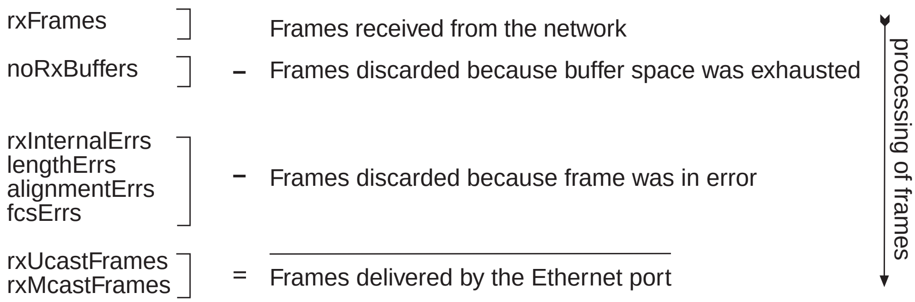
Ethernet Packet Processing

All frames on an Ethernet network are received promiscuously by an Ethernet port. A port can discard frames for either of the following reasons:

- There is no buffer space available.
- The frame is in error.

Figure 1 shows the order in which frame discard tests are made.

Figure 1 How Frame Processing Affects Ethernet Receive Frame Statistics

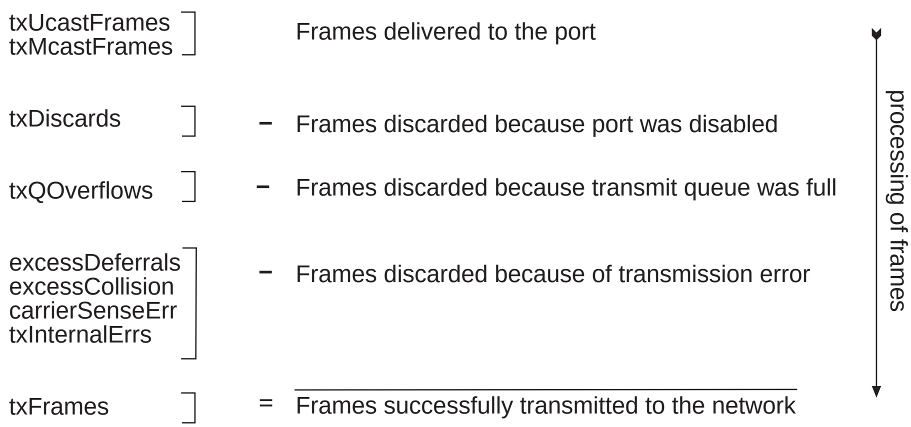


Frames also may be delivered directly to an Ethernet port by bridge, router, or management applications. However, a transmitted frame can be discarded for any of the following reasons:

- The Ethernet port is disabled.
- There is no room on the transmit queue.
- An error occurred during frame transmission.

Figure 2 shows the order in which these discard tests are made.

Figure 2 How Frame Processing Affects Ethernet Transmit Frame Statistics



Key Guidelines for Implementation

Consider these important factors when you implement and configure Ethernet networks:

Link Bandwidths

Recommended link capacities in a network normally depend on the speed requirements of end-user workstations, as shown in Table 35. In areas that may benefit from 1000-Mbps pipelines, you may be able to substitute trunked Fast Ethernet, subject to the issues raised in Chapter 12.

Table 35 Recommendations for Structuring Bandwidth Across the LAN

	Desktops to Wiring Closet	Wiring Closet to Data Center	Data Center to Campus Interconnect
Mainstream networks	Switched 10 or Shared 10/100	Switched 100	Switched 1000
Power networks	Switched 10/100	Switched 1000	Switched 1000+

Trunks

Consider these important factors when you implement and trunk Fast Ethernet or Gigabit Ethernet links:

- 3Com recommends that you use trunks to increase network availability in the following circumstances:
  - Switch-to-switch connections in the data center and campus interconnect areas
  - Switch-to-server connections in the data center and campus interconnect areas
  - Downlinks from the data center to the campus interconnect area
- When multiple links are trunked, it can be difficult to manage and troubleshoot individual port-to-port connections if a connectivity problem occurs. This issue may not be of concern in a server farm room. But if you use trunking extensively between wiring closets and data centers, the large number of connections involved and their distributed nature may make their management and troubleshooting difficult.

When working with trunks, be sure that you understand the port numbering for your system. For port-numbering information, see Chapter 7. For more information about trunking, see Chapter 12.

Port Enable and Disable (Port State)

You can `enable` Ethernet ports (place them on-line) or `disable` them (place them off-line).

Important Considerations

- Because it stops all network traffic through the port, disabling a port may adversely affect a live network.
- When a port is enabled, the port transmits frames normally. When a port is disabled, the port neither sends nor receives frames.
- The portState is `off-line` for disabled ports and `on-line` for enabled ports with an active link.

Port Labels

Port labels serve as useful reference points and as an accurate way to identify ports for management applications.

Implementing Port Labels

- Label Ethernet ports so that you can easily identify the devices that are attached to them (such as LANs, workstations, or servers). For example, you can assign `engineeringserver` as a label.
- A new port label appears in system displays the next time that you display information for that port.
- Port labels can include up to 32 ASCII characters, including the null terminator.

Autonegotiation

This feature enables some ports to identify and negotiate speed and duplex mode with a remote device.

Important Considerations

- In most cases, if autonegotiation does not properly detect the remote port speed, the vendor of the remote device implemented either autonegotiation or a change in port speed in a noncompliant way. If autonegotiation does not properly detect the port speed, you can manually set the port speed and duplex mode.
- Table 36 lists Ethernet port types on your system, whether they support autonegotiation, and which features they negotiate.

Table 36 Port Types and Autonegotiation Attributes

Port Type	Supports Autonegotiation	Negotiable Attributes	Default Values for Negotiable Attributes
10/100BASE-TX	Yes	Port speed	10 Mbps
		Duplex mode	Half-duplex
100BASE-FX	No	Not applicable	Not applicable
1000BASE-SX	Yes	Duplex mode	Full-duplex
		Flow control	If autonegotiation is enabled, the system's best effort is On
1000BASE-LX GBIC	Yes	Duplex mode*	Full-duplex*
		Flow control	If autonegotiation is enabled, the system's best effort is On
1000BASE-SX GBIC	Yes	Duplex mode*	Full-duplex*
		Flow control	If autonegotiation is enabled, the system's best effort is On
1000-Mbps backplane	No	Not applicable	Not applicable

* LX GBIC, and SX GBIC duplex modes are fixed at full-duplex at this release.

- **10/100BASE-TX ports** — Enabling autonegotiation causes both the port speed and duplex mode attributes to be autonegotiated.
- **100BASE-FX ports** — No autonegotiation of duplex mode occurs. The port speed is fixed at 100 Mbps. The default duplex mode is `half-duplex`.
- **1000BASE-SX ports** — Both link partners must either enable or disable autonegotiation. As long as autonegotiation is enabled, the system's best effort for handling flow control is `On`.

  When you enable autonegotiation, the system ignores your requested portMode information for 10/100BASE-TX ports and your requested flowControl information for 1000BASE-SX ports. When you disable autonegotiation, the system recognizes the requested portMode values for ports that have portMode options and the requested flowControl values for 1000BASE-SX ports. (Backplane ports do not support autonegotiation.)
- Use the `portMode` option to manually configure or modify the port speed and duplex mode. Use the `flowControl` option to manually configure or modify flow control.
- Autonegotiation is enabled by default on the ports that support it.

Port Mode

You can change the port speed and duplex mode for 10/100BASE-TX ports and the duplex mode for 100BASE-FX ports. You cannot change the port speed or duplex mode for Gigabit Ethernet ports.

Important Considerations

- When you configure duplex mode, configure both sending and receiving ports comparably. If the port speeds differ, the link does not come up. If the duplex modes differ, link errors occur.
- Table 37 lists the duplex port mode options available for each port type.

Table 37 Port Mode Options

Port Type	Duplex Port Mode	Resulting Port Mode	[Default]
10/100BASE-TX	100full	100 Mbps, full-duplex	10half
	100half	100 Mbps, half-duplex	
	10full	10 Mbps, full-duplex	
	10half	10 Mbps, half-duplex	
100BASE-FX	100full	100 Mbps, full-duplex	100half
	100half	100 Mbps, half-duplex	

- Enabling full-duplex mode on a port disables collision detection.
- Autonegotiation must be disabled on a port before a port mode selection can take effect.

Flow Control

The flow control mode allows a Fast Ethernet port or a Gigabit Ethernet port to:

- Decrease the frequency with which it sends packets to a receiving device, if packets are being sent too rapidly.
- Send flow control packets to a sending device, to request that the device slow its speed of transmission.

Important Considerations

Table 38 lists the effects of flow control options.

Table 38 Flow Control Options

Flow Control Option	Description	Available on Port Type
on	Port recognizes flow control packets and responds by pausing transmission. The port can generate flow control packets as necessary to slow incoming traffic.	Gigabit Ethernet Fast Ethernet
off	Port ignores flow control packets and does not generate flow control packets.	Gigabit Ethernet Fast Ethernet
rxOn	Port recognizes flow control packets and responds by halting transmission. The port does not generate flow control packets.	Gigabit Ethernet
txOn	Port ignores flow control packets, but it can generate flow control packets, if necessary.	Gigabit Ethernet

- The default setting for flow control is `off`.
- The system does not count flow control packets in receive or transmit statistics.

PACE Interactive Access

PACE Interactive Access (which is called PACE Access on Layer 2 modules) prevents excessive network jitter (variation in the timing of packet delivery that can cause garbled sound, jerky images, and delays). PACE technology also improves timing and optimizes LAN bandwidth utilization.

Important Considerations

- Use PACE Interactive Access only on half-duplex Ethernet links between a switch and a single end station (this setting has no effect on full-duplex links).
- Do not use PACE Interactive Access when a repeater is connected to a switch port.

Port Monitoring

The Ethernet port monitoring feature can prevent port duplex mismatches or excessive collisions from interfering with normal traffic forwarding. When enabled, this feature performs these functions:

- Monitors 10/100Mbps Ethernet ports for excessive collisions, multiple collisions, late collisions, runts, and FCS errors
- Compares these error counters against user-defined thresholds
- Disables a port that reaches an error threshold
- Reports the reason that a port is disabled to Switch 4007 management systems
- Reenables the port after an initial backoff time interval
- Continues monitoring

Standards, Protocols, and Related Reading

The system supports these Ethernet standards:

- **IEEE 802.3** — 10BASE-T Ethernet over unshielded twisted pair (UTP)
- **IEEE 802.3u** — 100BASE-T Fast Ethernet over UTP or fiber-optic cable
- **IEEE 802.3z** — 1000BASE-SX Gigabit Ethernet over multimode fiber-optic cable and 1000BASE-LX Gigabit Ethernet over multimode or single-mode fiber-optic cable

Ethernet Protocol

- **IEEE 802.3** — Carrier Sense Multiple Access with Collision Detection, which controls Ethernet access. A station that intends to transmit listens for network traffic. If it detects none, it transmits.

  If two or more stations transmit at about the same time, their packets experience a *collision* and the colliding data streams do not reach their destinations. The sending stations stop transmitting, broadcast a collision alert, and wait a random amount of time before trying again.

Media Specifications

Table 39 summarizes the system's Ethernet media options.

Table 39 Ethernet Media Specifications

Type	Speed	Media	Connector	Recommended Distance (max)
10/100BASE-TX	10/100 Mbps	Category 5 UTP	RJ-45	100 m
100BASE-FX	100 Mbps	single-mode fiber	SC	20 km
		multimode fiber	SC	412 m (half-duplex) 2 km (full-duplex)
1000BASE-SX	1000 Mbps	multimode fiber	SC	220 m (62.5 micron @ 160 MHz*km modal bandwidth)
				275 m (62.5 micron @ 200 MHz*km modal bandwidth)
				500 m (50 micron @ 400 MHz*km modal bandwidth)
				550 m (50 micron @ 500 MHz*km modal bandwidth)
1000BASE-LX GBIC	1000 Mbps	single-mode fiber	GBIC	5 km (9 micron) (qualified for up to 10 km)
		multimode fiber	GBIC, with duplex SC conditioned launch cable	550 m (62.5 and 50 micron @ all modal bandwidths)
1000BASE-SX GBIC	1000 Mbps	multimode fiber	GBIC	550 m (62.5 and 50 micron @ all modal bandwidths)

1000BASE Gigabit Interface Converter (GBIC) ports are hot-swappable, that is, you can replace one GBIC connector while the other connectors continue to carry traffic.

To ensure optimal compatibility, performance, and regulatory compliance, use only GBIC transceivers and conditioned launch cables that 3Com supports. For information about currently supported GBIC specifications and conditioned launch cables, see the 3Com Web site:

http://www.3com.com/gigabit_ethernet/gbics

Related Reading

For more information about Ethernet media options, see the *Switch 4007 Getting Started Guide*.

9 Bridge-Wide and Bridge Port Parameters

This chapter provides an overview of bridging concepts and implementation guidelines for modules in the Switch 4007. The chapter covers these topics:

- Bridging Overview
- Key Bridging Concepts
- Bridging Implementation Summary
- Key Guidelines for Implementation
- STP Terms and Concepts
- STP Bridge and Port Parameters
- MAC Address Table Design
- Address Aging
- Frame Processing
- IP Fragmentation
- IPX SNAP Translation
- Broadcast and Multicast Limits
- GARP VLAN Registration Protocol (GVRP)
- Standards, Protocols, and Related Reading

You can manage most bridge-wide and bridge port commands in either of these ways:

- *From the* `bridge` *menu of the Administration Console after you log in to the Management Module and connect to the module's slot.*
- *From the Bridge folder of the Web Management software.*

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Bridging Overview

Operating at the data link layer (Layer 2) of the OSI reference model, a bridge interconnects two or more LANs and allows them to communicate as if they were one LAN. Bridges examine incoming frames, make forwarding decisions from comparing the address information in the frame against the bridge's own address table as well as considering other factors such as VLANs.

Benefits

Bridges provide the following benefits:

- Bridges overcome cabling limitations and extend the effective length of a LAN, allowing distant stations to communicate that would not otherwise not.
- Bridges can provide a level of separation that prevents some potential damaging errors or undesirable frames from spreading or multiplying on the network.
- Because bridges only forward a percentage of total traffic received, they diminish the traffic that devices on connected segments experience and increase available bandwidth in each LAN.
- Bridges allow a larger number of devices to communicate than a single LAN can support.
- Bridges can detect loops in the network topology and communicate with each other to ensure that only one path exists between any two points and thus minimize the probability of broadcast storms.

In this chapter, the term bridge *refers to bridging operations on an individual switching module.*

The Switch 4007 software and management interfaces are built from CoreBuilder 9000 switch technology. In releases 3.0.0 and 3.0.5, the prompts and displays in all interfaces may indicate this heritage.

Key Bridging Concepts

Before you configure bridge-wide or bridge port settings on your module, you may find it helpful to review the following concepts.

Learning Addresses

Bridges *learn* addresses so that they can, over time, more effectively determine which frames to forward on which ports. A bridge learns addresses simply by processing the network traffic that it receives. For a bridge to learn the address of a station on the network that station must transmit a frame. The bridge learns the *source address* in the frame, not the destination address, and records it with the port on which the frame arrived. The next time that the bridge receives a frame that contains that learned address as its destination address, the bridge knows exactly to which port it should forward the frame.

Learned addresses that are called *dynamic addresses*, because they are learned on the fly and no human intervention is required. This term contrasts with *static addresses*, which are addresses that are manually configured.

A bridge maintains a database, called the *address table*, which lists all static and dynamic addresses and associates them with appropriate port numbers.

For more information about the module address tables, see "MAC Address Table Design" later in this chapter.

Aging Addresses

A dynamic address remains in the bridge's address table as long as the station to which it relates regularly transmits frames through the bridge. If the station does not transmit within a specified period of time, the source address is deleted, or *aged out*, from the table. Accelerated address aging may also occur if ports become disabled.

The process of aging addresses in combination with learning addresses ensures that, if a station moves to a different segment on the network or if the network topology changes, frames will continue to be forwarded correctly. Address aging is also necessary because a bridge can store only a finite number of addresses in its memory.

For information about how aging works in your modules, see "Address Aging" later in this chapter.

Forwarding, Filtering, and Flooding

A bridge filters, floods, or forwards frames by comparing:

- The frame's destination address to the source addresses in the bridge's address table.
- The destination bridge port (if known) to the port on which the frame was received.

The bridge compares the frame's destination address to the addresses in the address table and does one of the following:

- *If the destination address is known* to the bridge, the bridge identifies the port on which the destination address is located.
  - If the destination bridge port is *different* from the bridge port on which the frame was received, the bridge forwards the frame to the destination bridge port.
  - If the destination bridge port is the *same* as the port on which the frame was received, the bridge *filters* (discards) the frame.
- *If the destination address is not known* to the bridge, the bridge forwards the frame to all active bridge ports other than the bridge port on which the frame was received. This process is called *flooding*.

Other factors such as VLANs also affect how a bridge processes frames.

Loop Detection and Network Resiliency

To operate most efficiently, your network topology should have only one active path between any two bridging devices at any given time. When a bridge attaches to any single LAN with more than one active path, the network topology now has a *loop*.

When the bridge receives the same frame from multiple ports within a short period of time, a loop can cause a bridge to continually question where the source of a given frame is located. As a result, the bridge forwards and multiplies the same frame continually, which clogs up LAN bandwidth and challenges the processing capabilities of all devices on the LAN. This phenomenon of congestion, which can sometimes be so severe as to bring down network devices and network service, is called a *broadcast storm*.

You can configure a single path topology purely with cabling. However, the Spanning Tree Protocol (STP) gives you a way to configure *redundant* cable paths but yet maintain only one active path between two devices. STP monitors the status of all paths and, if an active path goes down, STP activates a redundant path and reconfigures the network topology accordingly. STP operations are fully described in the *IEEE 802.1D MAC Bridges* standard.

For more detailed information about Spanning Tree and the settings in your modules, see "STP Terms and Concepts" and "STP Bridge and Port Parameters" later in this chapter.

There are other ways to build redundancy into your network with or without using STP. Discuss your options with your network designer or 3Com product and service vendor.

Bridging Implementation Summary

Your module supports several features that relate to the bridging process and are therefore organized under the `bridge` menu on the interface.

The following features are covered in this chapter:

- **Compliance with IEEE 802.1D MAC Bridges standard** — Modules comply with the requirements that are outlined in the IEEE 802.1D Media Access Control (MAC) Bridges standard. Each module:
  - Supports transparent bridging, a form of bridging that listens promiscuously to every frame that is transmitted
  - Stores each received frame until the frame can be forwarded to one or more LANs or filtered.
  - Builds one or more address tables (depending on VLAN mode) by dynamically learning addresses as well as storing manually configured static addresses. See "MAC Address Table Design" in this chapter for more information.
  - Ages addresses out of the address table within defined periods of time if those stations have not transmitted frames or if ports have been disabled. See "Address Aging" in this chapter for more information.
  - Can detect loops in the network topology.

- **Spanning Tree Protocol** — You can configure bridge-wide and bridge port settings to enable STP to detect loops and calculate a network topology that reflects a single, loop-free path between any two devices. For conceptual information about STP, see "STP Terms and Concepts" in this chapter. For information on how to manipulate STP settings, see "STP Bridge and Port Parameters" in this chapter.
- **Multicast and broadcast limits** — You can assign per-port thresholds to limit the per-second forwarding rate of incoming broadcast and multicast traffic. See "Broadcast and Multicast Limits" in this chapter.
- **IP fragmentation (Multilayer Switching Modules only)** — When Fiber Distributed Data Interface (FDDI) stations transmit IP packets that are too large for standard Ethernet to handle, IP fragmentation allows a module to reformat large packets into smaller sizes. See "IP Fragmentation" in this chapter for more information.
- **IPX SNAP translation (Multilayer Switching Modules only)** — IPX SNAP Translation allows any 802.3_RAW IPX packets that are forwarded from Ethernet to FDDI to be translated to FDDI_SNAP (instead of FDDI_RAW), and vice versa. See "IPX SNAP Translation" in this chapter for more information.
- **GARP VLAN Registration Protocol (Multilayer Switching Modules only)** — GVRP simplifies the management of IEEE 802.1Q VLAN configurations in large networks by making aspects of VLAN configuration dynamic. See "GARP VLAN Registration Protocol (GVRP)" in this chapter, as well as the VLAN chapter in this guide.

The following features under the `bridge` menu are covered in other chapters in this guide:

- **Class of Service (Layer 2 Switching Modules only)** — A module can process frames through two priority queues. You assign each of the eight priority levels specified in the IEEE 802.1p standard to one of the two queues. For more information, see the Class of Service chapter in this guide.
- **Multicast filtering with IGMP (Layer 2 Switching Modules only)** — By understanding the Internet Group Management Protocol (IGMP), a module can direct IP multicast packets only to the ports that require them, instead of flooding to all ports. This process conserves bandwidth at the edge of the network. For more information, see the Multicast Filtering with IGMP chapter in this guide.

- **Resilient links (Layer 2 Switching Modules only)** — Resilient links protect your network against an individual link or device failure by providing a secondary backup link that is inactive until needed. For more information about resilient links, see the Resilient Links chapter in this guide.
- **Virtual LANs** — A Virtual LAN (VLAN) is a logical grouping methodology that allows dispersed users to communicate as if they were physically connected to the same LAN (broadcast domain). For more information, see the VLAN chapter in this guide.
- **Trunking** — You can aggregate multiple network links into a single point-to-point trunk. These features allow you to increase bandwidth and redundancy without replacing products or cabling. For more information, see the Trunking chapter in this guide.

Key Guidelines for Implementation

This section highlights the major issues to consider when you are planning how to configure bridging options on a module.

Additional, more specific guidelines are included in various sections throughout this chapter, usually under the heading "Important Considerations."

Physical Ports and Bridge Ports

All front-panel ports as well as backplane ports on all modules operate as bridge ports.

Option For Fast Aging

Even if you do not want to enable STP, you may need to configure a setting on the STP menu to activate accelerated aging when ports go down or are disabled. See "Address Aging" in this chapter for more information.

If You Want To Use STP

To function, STP must be enabled both on a bridge-wide basis and on a per-port basis. These are two different interface settings. If you disable bridge-wide STP, no ports on the module can participate in the STP algorithms for loop detection, even if STP is enabled on those bridge ports. STP is enabled by factory default on the bridge and for all bridge ports.

Port Forwarding Behavior

Table 40 summarizes the forwarding behavior of bridge ports based on the bridge and port STP states:

Table 40 Port Forwarding Behavior Depends on Bridge and Port STP States

Bridge STP State	Port STP State	Port Participates in STP?	Port Forwards Frames?
Disabled	Disabled	No	Yes, if link state is up.
	Enabled	No	Yes, if link state is up.
	Removed	No	Yes, if link state is up.
Enabled	Disabled	No	No
	Enabled	Yes	Determined by STP provided that the link state is up.
	Removed	No	Yes, if link state is up.

When STP is removed from the port but is enabled for the bridge, the port is invisible to STP but can forward frames. Removing the port from STP is useful if you have an edge switch device that is connected to end stations (such as PCs) that are frequently turned on and off.

Routing Over Blocked STP Ports

On a Multilayer Switching Module, you can elect to route certain specified traffic and bridge (using STP) other traffic simultaneously. To route packets from ports that are blocked by STP, you can configure a VLAN port to *ignore* the STP mode. For more information, see the VLAN chapter in this guide or see the command `bridge vlan stpMode` in the *Switch 4007 Command Reference Guide*.

STP Compatible with Trunking

You can enable STP on the same module and ports on which you configure trunks. STP understands that a trunk is one logical bridge port. In fact, you may find it useful to configure a backup trunk that STP places in the blocking state. Of course, a trunk itself has resilient properties in point-to-point connections, thus even if you do not enable STP, you can still gain a measure of resiliency in your network topology. See the Trunking chapter in this guide for more information.

STP Not Compatible with Resilient Links

If you want to define one or more resilient link pairs on a Layer 2 Switching Module, STP cannot be enabled:

- If STP is enabled and you define a resilient link pair, the module rejects it toward the end of the definition process.
- If you have one or more resilient links defined (STP is disabled) and you try to enable STP, the module rejects this request. You cannot enable STP until you remove all resilient link pairs.

Bridge Ports and Trunks

When you are prompted to select ports, you can specify the `?` option to see a matrix of information about your bridge ports, including a Selection column, a Port column, and a Label column.

- *Without trunks configured,* the Selection and Port columns contain the same port numbers, which indicates that you can select each port.
- *With trunks configured,* the Selection column indicates that you can select the anchor port (lowest-numbered port) in the trunk, and the Port column shows each port that is associated with the trunk. The Label column contains the trunk name, if you have assigned one.

Multicast Limits and Trunks

If you want to specify a multicast limit for a trunk, be sure to apply it to the trunk's anchor port (lowest-numbered port) only. However, be aware that the multicast limit will operate on *each port* in the trunk, even though you did not actively configure it that way. For example, if you have a trunk that consists of four ports and if you configure the anchor port with a multicast limit of 3000 pps, each of the four ports will operate with a 3000 pps limit.

Bridge Port Addresses in Closed VLAN Mode

If you are using `allClosed` for the VLAN mode and you want to administer bridge port address options in some way, be sure that you specify the correct VLAN interface index, because each VLAN in `allClosed` mode operates with a unique address table.

GVRP Usefulness

Configuring GARP VLAN Registration Protocol (GVRP) on Multilayer Switching Modules is useful only when there are several other switches or endstation NICs in the network that also support GVRP.

STP Terms and Concepts

This section provides a conceptual review of STP terms, parameters, and processes. To learn how you can manipulate STP parameters, see "STP Bridge and Port Parameters" in this chapter.

Configuration Messages

In order to determine a loopless LAN topology, bridges that support STP must communicate and share information with each other. STP Bridges periodically transmit special frames called *Configuration Bridge Protocol Data Units* (CBPDUs), which contain several pieces of information that help to determine the LAN topology. CBPDUs are stored in bridge memory but are refreshed periodically with the latest information.

STP uses an algorithm that compares the information from different CBPDUs to determine all possible paths and dynamically map out a loopless network topology. STP keeps one bridge port active and puts redundant bridge ports in the *blocking* state. A port in the blocking state neither forwards nor receives data frames.

After STP logically eliminates the redundant paths, the network configuration stabilizes. Thereafter, if one or more of the bridges or communication paths in the stable topology fail, STP recognizes the changed configuration and, within a few seconds, consults the stored CBPDU information and activates an appropriate number of redundant links to ensure that network connectivity is maintained.

CBPDUs do not propagate through the bridge as regular data frames do. Instead, each bridge acts as an end station — receiving, interpreting, and acting on the information in the CBPDUs.

Bridge Hierarchy

The CBPDUs help bridges establish a hierarchy (or a *calling order*) among themselves for the purposes of creating a loopless network.

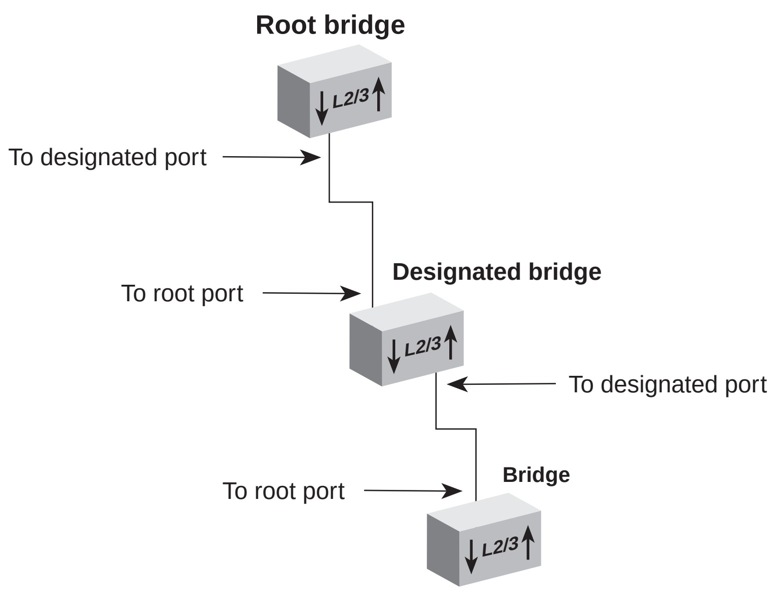
Based on the information in the CBPDUs, the bridges elect a *root bridge*, which is at the top level of the hierarchy. The bridges then choose the best path on which to transmit information to the root bridge.

The bridges that are chosen as the best path, called *designated bridges*, form the second level of the hierarchy.

- A designated bridge relays network transmissions to the root bridge through its *root port*. Any port that transmits to the root bridge is a root port.
- The designated bridges also have *designated ports* — the ports that are attached to the LANs from which the bridge is receiving information.

Figure 3 shows the hierarchy of the STP bridges and their ports.

Figure 3 Hierarchy of the Root Bridge and the Designated Bridge



Actions That Result from CBPDU Information

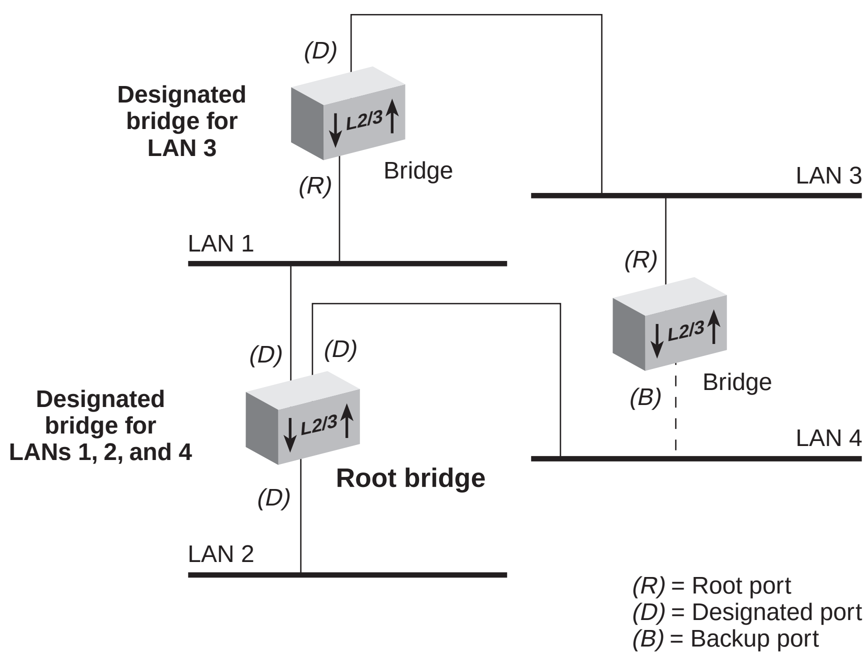
From the information that the CBPDUs provide:

- Bridges elect a single bridge to be the *root bridge*. The root bridge has the lowest bridge ID among all the bridges on the extended network.
- Bridges calculate the best path between themselves and the root bridge.

- Bridges elect as the *designated bridge* on each LAN the bridge with the *least cost path* to the root bridge. The designated bridge forwards frames between that LAN and the path to the root bridge. For this reason, the root bridge is *always* the designated bridge for its attached LANs. The port through which the designated bridge is attached to the LAN is elected the *designated port.*
- Bridges choose a *root port* that gives the best path from themselves to the root bridge.
- Bridges select ports to include in the STP topology. The ports that are selected include the root port plus any designated ports. Data traffic is forwarded to and from ports that have been selected in the STP topology.

Figure 4 shows a bridged network with its STP elements.

Figure 4 STP Root and Designated Bridges and Ports



Contents of CBPDUs

Bridges use information in CBPDU to calculate a STP topology. The content of a CBPDU includes:

- **Root ID** — The identification number of the root bridge.
- **Cost** — The cost of the least-cost path to the root from the transmitting bridge. One of the determining factors in cost is the speed of the bridge's network interface; that is, the faster the speed, the lower the cost.
- **Transmitting bridge ID** — The identification of the bridge that transmits the CBPDU, which includes the bridge address and the bridge priority.
- **Port identifier** — Includes the port priority as well as the number of the port from which the transmitting bridge sent the CBPDU.

  The port identifier is used in the STP calculation only if the root IDs, transmitting bridge IDs, and costs (when compared) are equal. In other words, the port identifier is a tiebreaker in which the lowest port identifier takes priority. This identifier is used primarily for selecting the preferred port when two ports of a bridge are attached to the same LAN or when two routes are available from the bridge to the root bridge.

Comparing CBPDUs

Here are three examples that show how the bridge determines the best CBPDU. In every case, the root ID is the most important determining factor. If the root ID fields are equal, then the cost is compared. The last determining factor is the transmitting bridge ID. If the CBPDUs all have the same root ID, cost, and transmitting bridge ID, then the port identifier is used as a tiebreaker.

Example 1. Root ID is lower for Message 1. The bridge saves Message 1.

Message 1			Message 2		
root ID	**cost**	**transmitter**	**root ID**	**cost**	**transmitter**
12	15	35	31	12	32

Example 2. Root ID is the same for Message 1 and Message 2, but cost is lower in Message 1. The bridge saves Message 1.

Message 1			**Message 2**		
root ID	**cost**	**transmitter**	**root ID**	**cost**	**transmitter**
29	15	80	29	18	38

Example 3. Root ID and cost are the same for Message 1 and Message 2, but the transmitting bridge ID is lower in Message 1. The bridge saves Message 1.

Message 1			**Message 2**		
root ID	**cost**	**transmitter**	**root ID**	**cost**	**transmitter**
35	80	39	35	80	40

How a Single Bridge Interprets CBPDUs

The following case describes how *a single bridge* interprets CBPDUs and contributes to the Spanning Tree configuration.

1 When Spanning Tree is first started on a network, the bridge acts as a root bridge and transmits a CBPDU from each of its ports with the following information:

- Its own bridge ID as the root ID (for example, `85`)
- Zero (0) as the cost (because, for the moment, it is the root bridge)
- Its own bridge ID as the transmitting ID (for example, `85`)

Thus, its CBPDU looks like this: `85.0.85`.

2 The bridge receives CBPDUs on each of its ports from all other bridges and saves the *best* CBPDU from each port.

The bridge determines the best CBPDU by comparing the information in each message that arrives at a particular port to the message that is currently stored at that port. In general, the lower the value of the CBPDU, the *better* it is. When the bridge comes across a better CBPDU than it has stored, it replaces the old message with the new one.

3 From the messages that are received, the bridge identifies the root bridge.

For example, if the bridge receives a CPBDU with the contents 52.0.52, then it assumes that the bridge with ID 52 is the root (because 52 is smaller than 85).

4 Because the bridge now knows the root bridge, it can determine its distance to the root and elect a root port.

It examines CBPDUs from all ports to see which port has received a CBPDU with the smallest cost to the root. This port becomes the root port.

5 Now that the bridge knows the contents of its own CBPDU, it can compare this updated CBPDU with the ones that its other ports received.

- If the bridge's message is better than the ones received on any of its ports, then the bridge assumes that it is the designated bridge for the attached LANs.
- If the bridge receives a better CBPDU on a port than the message it would transmit, it no longer transmits CBPDUs on that LAN. When the algorithm stabilizes, only the designated bridge transmits CBPDUs on that LAN.

How Multiple Bridges Interpret CBPDUs

The previous section looked at how a single bridge reviews CBPDUs and makes decisions. The following examples illustrate how STP determines the topology for an entire network.

Figure 5 and Figure 6 shows the same network topology — six bridges that connect six LANs. The topology is designed with redundant links for backup purposes, which create loops in the extended network. Figure 5 shows the network at the start of the STP topology calculation. Figure 6 shows the network after the STP topology has stabilized.

Figure 5 Starting the Spanning Tree Calculation

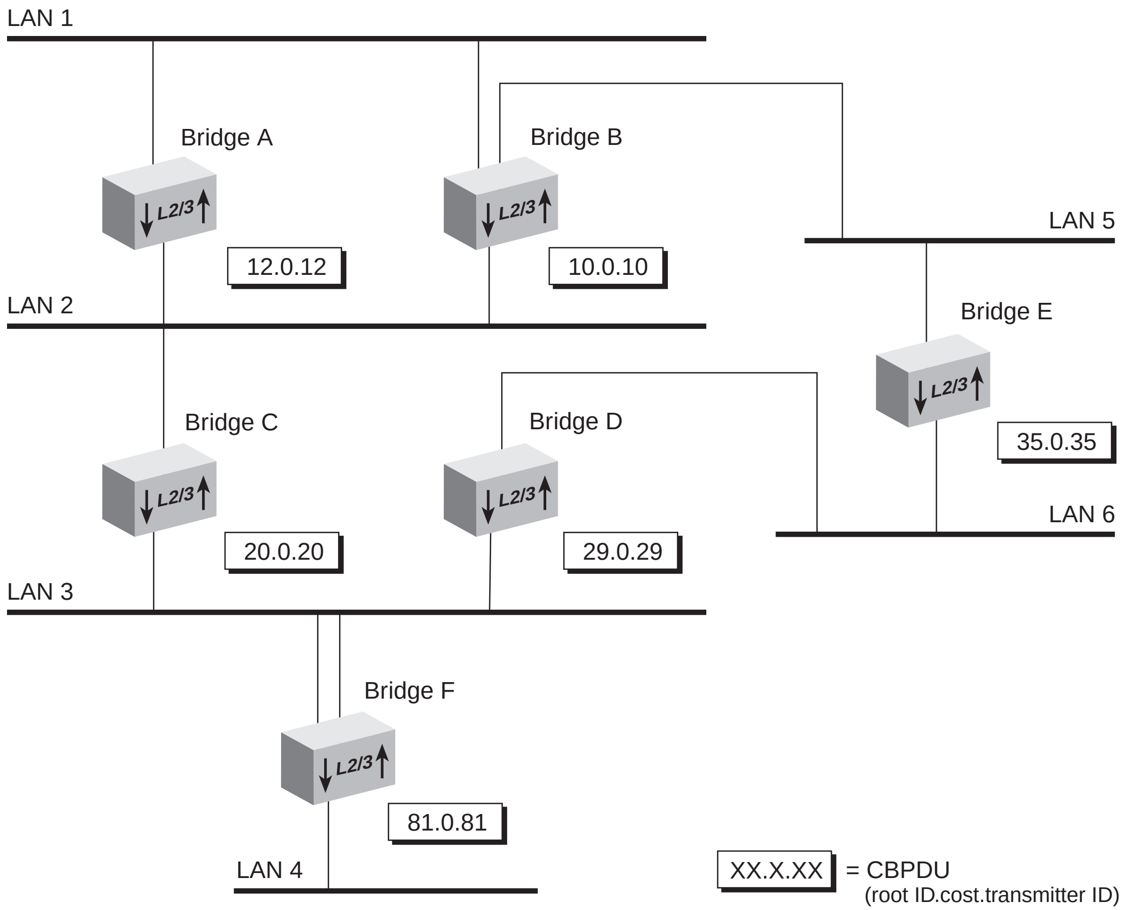
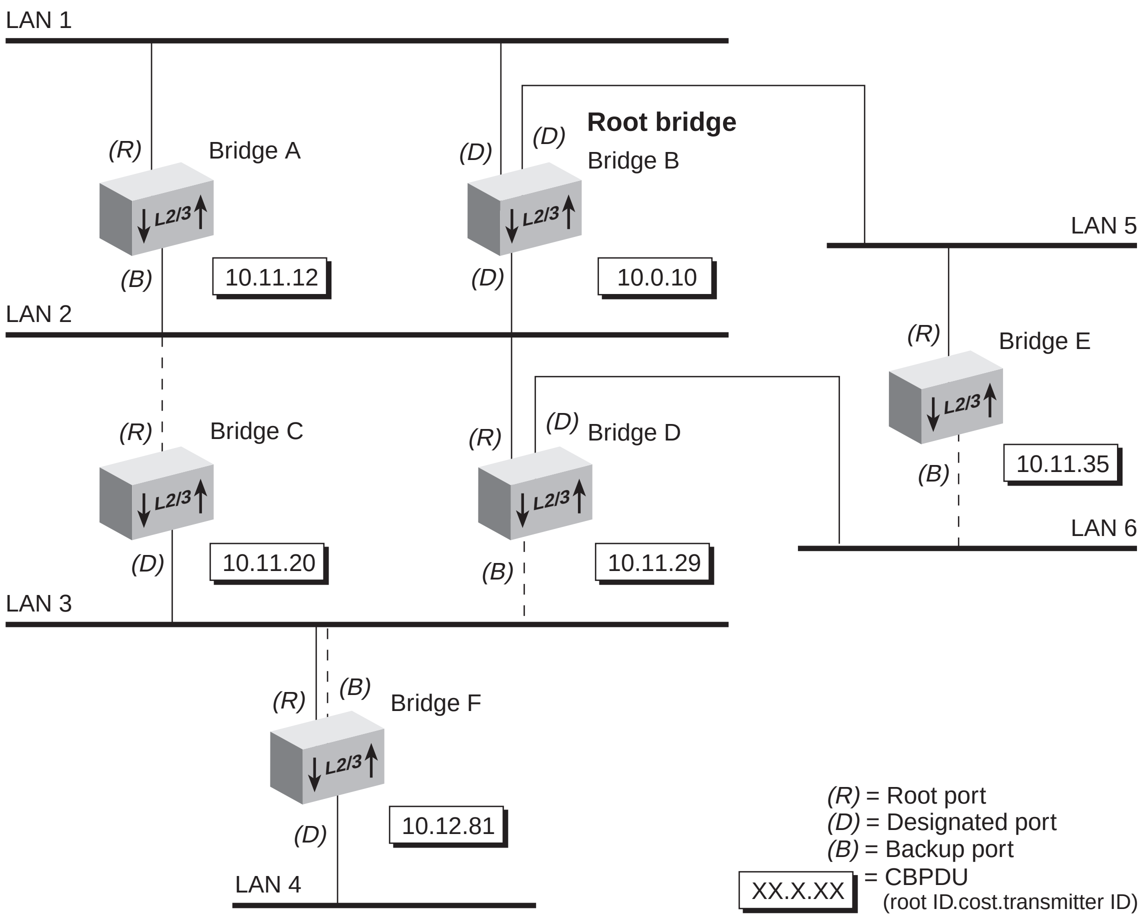


Figure 6 Spanning Tree Topology Calculated



Determining the Root Bridge

The root ID portion of the CBPDU determines which bridge actually becomes the root bridge. In Figure 5, notice how each bridge initially assumes that it is the root bridge and transmits a CBPDU that contains its own bridge ID as both the *root ID* and the *transmitting bridge ID* as well as a *zero cost*. In Figure 6, because Bridge B has the lowest root ID of all the bridges, it becomes the root and all other bridges change their root ID to Bridge B's ID (10).

Determining the Root Ports

Next, each bridge (except for the root bridge) must select a root port. To select a root port, each bridge determines the most cost-effective path for frames to travel from each of its ports to the root bridge. The cost depends on:

- The port path cost
- The root path cost of the designated bridge for the LAN to which this port is attached

If the bridge has more than one port attachment, the port with the lowest cost becomes the root port, and the other ports become either designated or backup ports. If bridges have redundant links to the same LAN, then the port with the lowest port identifier becomes the root port.

In Figure 6, Bridge F has two links to LAN 3 (through port 1 and port 2). Because the lowest port identifier for Bridge F is port 1, it becomes the root port, and port 2 becomes a backup port to LAN 3.

Determining the Designated Bridge and Designated Ports

For a LAN attached to a single bridge, that bridge is the LAN's designated bridge. For a LAN that is attached to more than one bridge, a designated bridge must be selected from among the attached bridges.

The root bridge is automatically the designated bridge for all of its directly attached LANs.

For example, Bridge B, the root bridge in Figure 6, is also the designated bridge for LANs 1, 2, and 5.

A designated bridge must be determined for LANs 3, 4, and 6:

- Because Bridges C, D, and F are all attached to LAN 3, one of them must be the designated bridge for that LAN:
  - The algorithm first compares the root ID of these bridges, which is the same for all.
  - The cost is then compared. Bridge C and Bridge D both have a cost of 11. Bridge F, with a cost of 12, is eliminated as the designated bridge.
  - The transmitting bridge ID is compared between Bridge C and Bridge D. Because Bridge C's ID (20) is smaller than Bridge D's (29), Bridge C becomes the designated bridge for LAN 3.
- The designated bridge for LAN 6 is either Bridge D or Bridge E. Because Bridge D's transmitting bridge ID (29) is lower than Bridge E's (35), Bridge D becomes the designated bridge for that LAN.
- The designated bridge for LAN 4 is Bridge F, the only bridge that is attached to that LAN.

The port that attaches the designated bridge to the LAN determines the designated port. If more than one port is attached to the same LAN, then the port identifier determines the designated port.

Spanning Tree Port States

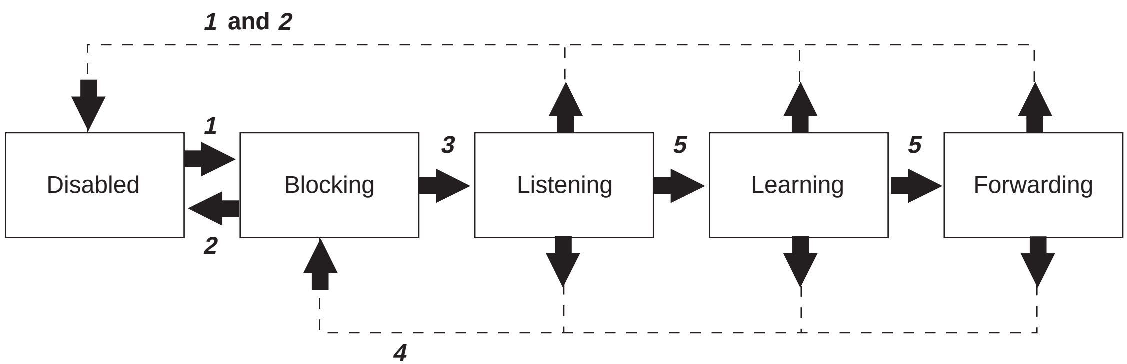
Because STP determines the network configuration based on events that occur, it places bridge ports in one of the five states at all times. Table 41 describes these states.

Table 41 Spanning Tree Protocol Port States

Port State	Description
Listening	When STP is configuring, all ports are placed in the listening state. Each port remains in this state until the root bridge is elected. While in the listening state, the bridge continues to run STP and to transmit CBPDUs on the port; however, the bridge discards data frames that are received on that port and does not transmit data frames from that port. The listening state should be long enough for a bridge to hear from all other bridges on the network. After being in the listening state, the bridge ports that are to proceed to the forwarding state go into the learning state. All other bridge ports go into the blocking state.
Learning	The learning state is similar to the listening state except that data frames are received on that port for the purpose of learning which stations are attached to that port. After spending the specified time in this state without receiving information to change the port back to the blocking state, the bridge changes the port to the forwarding state. The time that the port spends in each of the listening and learning states is determined by the value of the *forward delay* parameter.
Forwarding	After the port enters the forwarding state, the bridge performs standard bridging functions.
Blocking	When a port is put in the blocking state, the bridge continues to receive CBPDUs on that port (monitoring for network reconfigurations), but it does not transmit them. In addition, the bridge does not receive data frames from the port, learn the locations of station addresses from it, or forward frames onto it.
Disabled	A port is disabled when the STP has been disabled on the port or when the port has failed. In the disabled state, the port does not participate in the Spanning Tree algorithm. The port continues to forward frames only if STP is disabled for the entire bridge and the link is up.

Figure 7 illustrates the factors that cause a port to change from one state to another. The arrows indicate the direction of movement between states. The numbers correspond to the factors that affect the transition.

Figure 7 Factors Involved in Spanning Tree Port State Transitions



1	Port enabled by either network administrator or initialization
2	Port disabled by either network administrator or failure
3	Spanning Tree algorithm selects port as designated or root
4	Spanning Tree algorithm does not select port as designated or root
5	Forwarding timer (forward delay) expires

As shown in Figure 7, for a port in the blocking state to transition to the listening state, STP must select that port as a designated or root port. After the port enters the listening state, the forward delay must expire before the port can transition to the learning state. Then another forward delay must expire before the port can transition to the forwarding state. If you disable a port in the listening, learning, or forwarding state or if port initialization fails, then that port becomes disabled.

Reconfiguring the Bridged Network Topology

STP reconfigures the bridged network topology when any of the following occurs:

- Bridges are added or removed.
- The root bridge fails.
- You change any of the bridging parameters that determine the topology.

Resulting Actions

Whenever a designated bridge detects a topology change, it sends a Topology Change Notification Bridge Protocol Data Unit (BPDU) through its root port. This information is eventually relayed to the root bridge.

The root bridge then sets the Topology Change Flag in its CBPDU so that the information is broadcast to all bridges. It transmits this CBPDU for a fixed amount of time to ensure that all bridges are informed of the topology change.

If a port changes from the blocking state to the forwarding state as a result of the topology change, STP sends the topology information to all the ports before that port starts forwarding data. This delay prevents temporary data loops.

When a network reconfiguration occurs, a bridge flushes all addresses from the address table. This ensures that a bridge learns the correct addresses and paths and continues to forward frames to the correct LAN.

STP Bridge and Port Parameters

On any switching module, if you want to use STP, you must first enable it on a bridge-wide basis and then on a per-port basis. This section describes the parameters that you can modify and their implications for your network.

On Layer 2 Switching Modules, even if you do not want to use STP, you may want to configure the `agingOnly` *option that exists in the* `bridge spanningtree stpState` *command — this setting relates to the module's ability to accelerate address aging in certain circumstances. See "MAC Address Table Design" and "Address Aging" in this chapter for more information.*

Bridge-wide STP Parameters

You can modify these STP bridge-wide parameter at any times:

- Bridge-wide STP state
- Bridge priority
- Bridge maximum age
- Bridge hello time
- STP group address

See the Switch 4007 Command Reference Guide *for value ranges and defaults for these parameters, as well as definitions of fields in the displays.*

Bridge-Wide STP State

You can set the bridge-wide STP state to one of these options:

Default

- **Enabled** — This setting allows any bridge port on the module to run STP, as long as it too has STP enabled. The bridge-wide STP state is enabled by default on all modules.
- **Disabled** — The module operates as a store-and-forward bridge, but cannot participate in the STP algorithm. Thus, the STP settings on individual bridge ports have no effect either.
- **Aging Only (Layer 2 Switching Modules only)** — STP is disabled with this setting, however an accelerated address aging process will occur when the module detects a port link down event. (This is a process that also occurs when STP is enabled.)

To understand how the bridge-wide STP state options affect address aging, see "Address Aging" in this chapter.

Bridge Priority

The *bridge priority* influences the choice of the root bridge and the designated bridge. The *lower* the bridge's priority number, the *more likely* it is that the bridge is chosen as the root bridge or a designated bridge. The bridge priority value is appended as the most significant portion of a bridge identifier. The factory default is 0x8000.

Bridge Maximum Age

The *maximum age* determines when the stored configuration message information is judged to be too old and is discarded from the bridge's memory. If the value is too small, then STP may reconfigure the topology too often, causing temporary loss of connectivity in the network. If the value is too large, the network may take longer than necessary to adjust to a new STP configuration after a topology change such as the restarting of a bridge. A conservative value assumes a delay variance of 2 seconds per hop. The recommended value (factory default) is 20 seconds.

Either the factory default value or a modified value that you set for maximum age on a given module is only used if the module is selected as the root bridge. Otherwise, the module uses the maximum age value that is assigned to it by the root bridge via the CBPDU. The `bridge display` shows both values — the one set by the root bridge and the one configured on the device.

Bridge Hello Time

Hello time is the period between the configuration messages that a root bridge generates. If the probability of losing configuration messages is high, shorten the time to make STP more robust. Alternatively, to lower the overhead of STP, lengthen the time. The recommended value (factory default) is 2 seconds.

Either the factory default value or a modified value that you set for hello time on a given module is only used if the module is selected as the root bridge. Otherwise, the module uses the value that is assigned to it by the root bridge via the CBPDU. The `bridge display` shows both values — the one set by the root bridge and the one configured on the device.

Bridge Forward Delay

The *forward delay* value specifies the amount of time that a bridge spends in each of the listening and the learning states. This value temporarily prevents a bridge from starting to forward data frames to and

from a link until news of a topology change has spread to all parts of a bridged network. The delay gives enough time to turn off to all links that need to be turned off in the new topology before new links are turned on.

Setting the value too low can result in temporary loops while STP reconfigures the topology. Setting the value too high can lead to a longer wait while the STP reconfigures the topology. The recommended value (factory default) is 15 seconds.

Either the factory default or a modified value that you set on a given module for bridge forward delay is only used if that module is elected as the root bridge. Otherwise, the module uses the value that is assigned to it by the root bridge via the CBPDU. The `bridge display` shows both values — the one set by the root bridge and the one configured on the device.

The forward delay value may also be used in the address aging process and may be used to delay link up and link down traps. See "Address Aging" later in this chapter for more information.

STP Group Address

The STP group address is a single address to which a bridge listens when it receives STP information. Each bridge on the network sends STP frames to the group address. Every bridge on the network receives STP frames that were sent to the group address, regardless of which bridge sent the frames.

You can run separate STP domains in your network by configuring different STP group addresses. A bridge only acts on STP frames that are sent to the group address for which it is configured. Frames with a different group address are ignored.

Because there is no absolute standard for the group address, products from different vendors may be configured with dissimilar group addresses as their default. If STP does not seem to be working in a mixed-vendor environment, view the group address on each device and change them to be identical if necessary.

Bridge Port STP Parameters

You can modify these STP parameters on each port:

- Port state
- Port path cost
- Port priority

Port State

You can enable, disable, or remove STP for each bridge port on a module. This setting affects the operation of a port only if bridge-wide STP is enabled. Table 42 summarizes the forwarding behavior of bridge ports based on the bridge-wide STP and per-port STP states:

Table 42 Port Forwarding Behavior Depends on Bridge and Port STP States

Bridge STP State	Port STP State	Port Participates in STP?	Port Forwards Frames?
Disabled	Disabled	No	Yes, if link state is up.
	Enabled	No	Yes, if link state is up.
	Removed	No	Yes, if link state is up.
Enabled	Disabled	No	No
	Enabled	Yes	Determined by STP provided that the port link state is up.
	Removed	No	Yes, if link state is up.

Port Path Cost

The algorithm adds the path cost to the root cost field in a configuration message that is received on this port. The module uses this value to determine the path cost to the root through this port. You can set this value individually on each port.

A higher path cost value makes the LAN that is reached through the port more likely to be low in the STP topology. The lower the LAN is in the topology, the less through traffic it carries. For this reason, assign a high path cost to a LAN that has a lower bandwidth or to one on which you want to minimize traffic.

Port Priority

The STP port priority influences the choice of port when the bridge has two ports connected to the same LAN, which creates a loop. The port with the lowest port priority is used by STP.

MAC Address Table Design

All modules recognize two different kinds of addresses:

- **Static MAC addresses** — Addresses that you manually add to the bridge address table using menu options. These addresses never age; you must add and remove them manually.
- **Dynamic MAC addresses** — Addresses that the bridge learns by receiving and processing frames and ages. In the bridge address table, each dynamic address is associated with a specific port and is assigned an age so that it can be cleared from the table if the station is inactive.

Among all Switch 4007 modules, the address table design, learning process, and capacity are identical. The aging process differs slightly between Layer 2 Switching Modules and Multilayer Switching Modules.

Address Space

Each module can each store up to 32,768 addresses (32K):

- In open VLAN mode, the address space is contiguous.
- In closed VLAN mode, the address table is dynamically allocated among the VLANs so that in effect each VLAN operates with its own address table.
- The *address threshold* (Multilayer Switching Modules only) is the value at which a module reports the total number of addresses that are known. Specifically, when this threshold is reached, the module generates the SNMP trap called `addressThresholdEvent`. The range of values you can enter for this parameter is between 1 and 1 plus the maximum address table size. If you do not want the module to generate events, set address threshold to one greater than the address table; the threshold will never be reached.

Important Considerations

The following address table considerations apply to all modules, unless otherwise specified or unless the functionality described does not exist on your module:

- The address table is flushed any time that you cycle power to a module or the chassis, reboot the module or chassis, or reset non-volatile data.
- Dynamic addresses are flushed whenever STP reconfigures the topology.

- You can remove individual MAC addresses from selected ports. Typically, this action is only applied to the removal of static addresses because the module can quickly relearn dynamic addresses that you remove.
- A static address is never aged from the address table and it cannot be learned dynamically on a different port until it is removed from the port on which it is configured as a static address.
- If a station whose address is statically-configured on one port is moved to a different port, the module discards all received frames as a security measure and increments a statistical counter. (From the `bridge display` of the Administration Console, see the `rxSecurityDiscs` field. From the Bridge Display option on the Web Management interface, see the Received Security Discards column.)
- The number of static MAC addresses that you can configure depends on the availability of module resources. You can configure up to 16 static addresses minimum.
- If you select closed VLAN mode, the address space is divided dynamically between the VLANs.
- The process of aging addresses differs slightly between Layer 2 Switching Modules and Multilayer Switching Modules. See "Address Aging" next in this chapter for more information.
- If you have multiple ports associated with a trunk, the addresses that are defined for the anchor port apply to all ports in the trunk.

Address Aging

This section explains how address aging works and identifies the slight differences in options and operation between Layer 2 Switching Modules and Multilayer Switching Modules.

Address aging only applies to dynamic addresses. (Static addresses are never aged from the address table and cannot be learned on other ports until they have been manually removed.)

Address Table Dependencies

The amount of time that dynamic addresses remain in a module's address table depends on these factors:

- The aging interval value that you configure (`bridge agingTime`)
- The option that you select for the bridge STP state:
  - For Multilayer Switching Modules, choose `enabled` or `disabled`
  - For Layer 2 Switching Modules, choose `enabled`, `disabled`, or `agingOnly`
  - These options are explained later in this section.
- The module's detection of link state changes on ports; shorter aging intervals may be applied for a certain period of time, depending on the STP state that you have selected.
- The value that you configure for STP forward delay on the module or that is assigned to the module by the STP root bridge. The value used depends on the STP state that you have selected.
- Other non-aging factors, such as STP topology changes and power cycles. See "Important Considerations" in the previous section "MAC Address Table Design" for more information.

On Layer 2 Switching Modules, if you do not *want to enable STP but you* do *want accelerated aging to occur when ports go down, select the* `agingOnly` *option from the* `bridge spanningTree stpState` *command. You may also want to modify the two STP parameters that the* `agingOnly` *function uses. None of these actions will enable STP itself.*

Normal Aging Process

Each module records in its address table the source address of every received frame (that is not otherwise filtered or discarded) along with the appropriate port number.

At the time the table entry is created, an aging bit for that entry is set to 1. When the entry is refreshed (i.e., the port receives a frame with the same address), the bit is changed to 0. At specific intervals, the module scans the address table for addresses that have bit values of 1 and removes or *ages out* those entries. Once the scan and removal process has finished, bits at 0 change to 1 so that if their entries are not refreshed by the time of the next scan, they will be aged out at that time.

Thus, it can take from less than one to two intervals to age out an address table entry under normal circumstances. From the module's Administration Console, you can modify the aging interval using the `bridge agingTime` command.

If the STP State is Enabled

When you set the `bridge spanningTree stpState` option to enabled and the module is operating in a stable STP network configuration, the module ages all dynamically learned addresses as described in "Normal Aging Process" (the preceding section). Two situations can affect this process:

- Module receives notice of an STP topology change
- Module detects a port down event

STP Topology Change

During the time that the root bridge sends CBPDUs with the topology change flag set, the module uses a different (usually shorter) interval to age addresses in accordance with STP rules:

If the module is operating as the root bridge, it uses the forward delay value that exists in its configuration. From the Administration Console, you can modify the value using the `bridge spanningTree stpForwardDelay` command. This value is shown in the `bridgeFwdDelay` field of the `bridge display`.

If the module is not operating as the root bridge, it uses the value that is assigned to it by the root bridge via the CBPDU. This value is shown in the `forwardDelay` field of the `bridge display`. To adjust this value, you must find and connect to the root bridge.

The module reverts to using the `bridge agingTime` value as the aging interval after it receives a CBPDU from the root bridge that does not have the topology change flag set.

Port Down Events

When a Multilayer Switching Module detects a port down event, it immediately flushes all addresses for that specific port. This process occurs no matter what setting exists for the bridge or port STP states.

When a Layer 2 Switching Module that has STP enabled detects a port down event, if that port is not an active STP port (in the forwarding state), there is no change to aging behavior. Otherwise, the normal STP process of temporarily using the forward delay value applies.

If the STP State is Disabled

In this state, the module does not participate in STP operations. The aging process works as follows:

- **Multilayer Switching Modules** — These modules age dynamic addresses as described in "Normal Aging Process" earlier in this section, except for when link down events are detected. When a port goes down or is disabled, the module immediately flushes all addresses for that specific port. (This process occurs no matter what setting exists for the bridge or port STP states.)
- **Layer 2 Switching Modules** — At all times, including when ports go down or are disabled, it ages dynamic addresses as described in "Normal Aging Process" earlier in this section.

  If you do not want to enable STP, but you want addresses to age faster when ports go down, use the `bridge spanningTree stpState agingOnly` option (described next).

If STP State is "Aging Only"

This option is available on Layer 2 Switching Modules only. With this option selected, the module does not operate as an STP bridge but does use two of the STP parameters in certain circumstances related to aging addresses.

During normal operation, address aging works as described in "Normal Aging Process" earlier in this section. But the module adjusts the aging interval when it detects down or disabled ports — it uses the value in the `bridge spanningTree stpForwardDelay` option as the aging interval.

In contrast to how Multilayer Switching Modules work, this aging process affects all dynamic addresses in the address table, not just the addresses that are associated with the ports that went down. Multilayer Switching Modules only flush addresses that are associated with specific ports.

A Layer 2 Switching Module reverts back to using the value in `bridge agingTime` after a period of time equal to `bridge spanningTree stpForwardDelay + bridge spanningTree stpMaxAge` seconds has transpired.

Important Considerations

The factory default values for the aging period (`bridge agingTime`) and for the STP forward delay (`bridge spanningTree stpForwardDelay`) align with the recommendations in the IEEE 802.1D specification. Here are some things to consider when adjusting these values:

- If you decrease the `bridge agingTime`, you may see increased levels of network traffic because address entries will age out faster and, upon receiving frames with unknown source addresses, the bridge must flood frames to all ports (except the one on which is was received) in accordance with IEEE 802.1D.
- If you increase the `bridge agingTime`, flooding may decrease but movement in edge devices may not be detected as quickly.
- If you decrease or increase the `bridge spanningTree stpForwardDelay` (shown as `bridgeFwdDelay` on the `bridge display`), it will only have an affect if:
  - The STP state is set to `agingOnly`, or
  - The STP state is set to `enabled` *and* the module is the root bridge.

- If the functioning forward delay is on the high end of the allowable range (4-30 seconds, 15 is the default), this allows more time for an STP network to stabilize but it also delays the transition of ports to the forwarding state because ports are held longer in each of the listening and learning states. However, when this value is used as the aging interval, the interval will probably still be much lower than the normal aging interval (range 10 - 1,000,000 seconds; 300 is the default).
- If the functioning forward delay is on the low end of the range and STP is enabled, ports may transition to the forwarding state before the network topology stabilizes, thereby potentially causing loops.
- If you select the `agingOnly` option on Layer 2 Switching Modules, 3Com recommends that you set the bridge spanningTree stpForwardDelay value to the lowest possible value — 4 seconds. This will ensure that the address flushing is performed as quickly as possible.
- If you select either `enabled` or `agingOnly` for the bridge-wide STP state on Layer 2 Switching Modules, the reporting of link up and down events is delayed by the value of the functioning forward delay. This delay helps ensure that the application Transcend Enterprise VLAN Manager (EVM) operates correctly.

  EVM associates a MAC address with each port that it assigns to a VLAN. If MAC addresses are not flushed out fast enough after a link up or link down event, the next EVM query may detect an old address on a given port and use it to determine VLAN membership for the port. The delay in sending traps allows sufficient time for addresses to be aged out before EVM queries arrive to retrieve the MAC address on a given port, thereby guaranteeing correct VLAN assignments.

The delay is not necessary on Multilayer Switching Modules because, irrespective of the STP state, the modules immediately flush dynamic addresses associated with down or disabled ports.

- The "Aging Only" option was created primarily for customers with Layer 2 Switching Modules who use 3Com Transcend EVM but do not want to enable STP. If you *do not* want to enable STP and if you *do not* want to accelerated address aging on link down events and if you *do* want traps reported immediately, then set the STP state to `disabled`.

Frame Processing

All frames that are received on a physical interface and that are not either discarded or explicitly directed to the module itself are delivered to the corresponding bridge port. The bridge port either forwards each frame to another port or discards it.

A module can discard an incoming frame for the following reasons:

- The destination station is on the same segment as the source station.
- The receive bridge port is blocked by STP.
- There is a problem with the frame.

  The physical interface does not deliver frames with errors to the bridge port. Thus, the `rxFrames` fields in the Ethernet statistics display and bridge statistics display often report different values — that is, the latter value is lower because it does not count frames in error.
- A user-defined frame filter (Multilayer Modules only) indicated not to forward the frame.
- The frame exceeded the configured broadcast or multicast rate limit

A frame that is forwarded to a bridge port is then transmitted to a physical interface unless it is discarded. A module can discard a frame at this point for the following reasons:

- The transmit bridge port is blocked.
- The frame is too large for the corresponding physical interface.
- A user-defined packet filter (Multilayer Modules only) indicated not to forward the frame.

IP Fragmentation

Standard FDDI allows larger maximum packet sizes than standard Ethernet. FDDI stations that transmit IP packet sizes larger than approximately 1500 bytes cannot communicate with stations on an Ethernet LAN. If a Layer 2 Switching Module receives such packets and they are destined for one or more Ethernet LANs, it filters them. The same happens with Multilayer Switching Modules, unless the IP fragmentation options is enabled.

When you enable IP fragmentation (Multilayer Switching Modules only), the module breaks up large FDDI packets that is receives into smaller packets before bridging them to Ethernet ports.

IPX SNAP Translation

IPX SNAP Translation (Multilayer Switching Modules only) allows an alternative method of translating IPX packets from Ethernet to FDDI and vice-versa.

- When IPX SNAP translation is enabled, any 802.3_RAW IPX packets that are forwarded from Ethernet to FDDI are translated to FDDI_SNAP. Likewise, SNAP IPX packets that are forwarded from FDDI to Ethernet are translated to 802.3_RAW packets.
- When IPX SNAP translation is disabled, the module uses standard IEEE 802.1H bridging to translate 802.3_RAW packets to FDDI_RAW packets.

Broadcast and Multicast Limits

You can assign a rate limit to any bridge port to control the per-second forwarding rate of incoming multicast and broadcast traffic. If the limit is reached, all remaining frames received in that second of time are dropped. This feature is useful for suppressing potential multicast or broadcast storms.

To set a bridge port limit:

1. Measure the normal broadcast and multicast traffic flow on the bridge port and determine an appropriate limit.
2. At the top-level module prompt, enter **`bridge port multicastLimit`**
3. Specify the port number
4. Specify the type of frame to which the limit should apply. Select either:
   - **`McastBcast`** (multicast and broadcast)
   - **`BcastOnly`** (broadcast only)

   This step pertains to Layer 2 Switching Modules only; on Multilayer Switching Modules, the limit always applies to both multicasts and broadcasts.
5. Specify a value for the rate limit
   - For Multilayer Switching Modules, the value you enter represents K frames per second (approximately 1000 fps). For example, if you enter 3, in practice it is 3000 fps.
   - For Layer 2 Switching Modules, the value you enter represents frames per second. For example, if you enter 3, in practice it is 3 fps.

Important Considerations

- If the limit that you set is reached during a given second of time, all remaining frames that are received in that remainder of that second are dropped.
- A value of zero means that there is no limit set on the port. This is the default setting for all ports.
- On Multilayer Switching Modules, the limit always affects both multicast and broadcast packets. You can set similar limits using options on the Quality of Service menu.
- If you want to specify a limit for a trunk, you only need to specify the trunk's anchor port (the lowest-numbered port). However, be aware that the limit operates on each link in the trunk, even though you only configured it on the anchor port.
- If you have IP multicast application traffic on your network, be sure that any limits that you configure do not constrain these traffic flows. On Layer 2 Switching Modules, if you select the `BcastOnly` option, IP multicast traffic is not affected.

GARP VLAN Registration Protocol (GVRP)

To activate GVRP on a Multilayer Switching Module, you enable the GARP VLAN Registration Protocol (GVRP) first on the bridge and then on individual bridge ports. On a port-by-port basis, GVRP allows the module to automatically learn the presence of and updates to 802.1Q VLANs. GVRP simplifies the management of IEEE 802.1Q VLAN configurations in large networks by making aspects of VLAN configuration dynamic.

GVRP maintains a database of VLAN member ports as the bridge learns about them. Specifically, GVRP tracks which ports are added to and removed from each VLAN and communicates this information to other GVRP-aware bridges. The bridges then determine active topologies for the network and for each VLAN using STP to prevent network loops.

GVRP operates only on ports that are in the STP forwarding state. If GVRP is enabled, a port that changes to the STP forwarding state automatically begins to participate in GVRP. A port that changes to an STP state other than forwarding no longer participates in GVRP.

Important Considerations

To use GVRP, consider the following:

- GVRP updates are not sent out to any blocked STP ports. GVRP operates only on ports that are in the STP forwarding state. If GVRP is enabled, a port that changes to the STP forwarding state automatically begins to participate in GVRP. A port that changes to an STP state other than forwarding no longer participates in GVRP.
- GVRP is disabled by default on the bridge and on all bridge ports.
- Enabling GVRP determines whether the VLAN origin for a port-based VLAN is dynamic (GVRP enabled) or static (GVRP disabled).
- To maximize the effectiveness of GVRP, it should be supported in as many end stations and network devices as possible.
- Based on updates from GVRP-enabled devices, GVRP allows the module to dynamically create a port-based VLAN (unspecified protocol) with a specific VLAN ID and a specific port.
- On a port-by-port basis, GVRP allows the module to learn about GVRP updates to an existing port-based, protocol-based, or network-based VLAN with that VLAN ID and IEEE 802.1Q tagging.
- VLANs that are created dynamically with GVRP exist only as long as a GVRP-enabled device is sending updates — if the devices no longer

send updates, or if GVRP is disabled, or if the module is rebooted, all dynamic VLANs are removed.

- GVRP manages the active topology, not nontopological data such as VLAN protocols. If a local bridge needs to classify and analyze frames by VLAN protocols, you must manually configure protocol-based VLANs and simply rely on GVRP to send VLAN ID updates. But if the local bridge needs to know only how to reach a given VLAN, then GVRP provides all necessary information.
- The VLAN topologies that GVRP learns are treated differently from VLANs that are statically configured. Although static updates are saved in nonvolatile RAM, GVRP's dynamic updates are not. When GVRP is disabled, the module deletes all VLAN interfaces that were learned through GVRP and leaves unchanged all VLANs that were configured through the Administration Console or through the Web management software.

Standards, Protocols, and Related Reading

Refer to the following standard for more information about the bridging methodology described in this chapter:

- **IEEE 802.1D** — This standard specifies requirements for transparent bridging.
- **IEEE 802.1Q** — This standard defines a new frame format, GVRP, and the dynamic registration of VLANs.

To obtain copies of these standards, register for an on-line subscription at the Institute of Electrical and Electronics Engineers (IEEE) Web site:

`http://www.ieee.org`

10 CLASS OF SERVICE (COS)

The *IEEE 802.1D Media Access Control (MAC) Bridges* standard has been amended in recent years to include various supplements. One such supplement standard is *IEEE 802.1p: Traffic Class Expediting and Dynamic Multicast Filtering*. This chapter describes the traffic prioritization portion of this standard and how it is implemented in Switch 4007 Layer 2 Switching Modules. This chapter covers these topics:

- Overview
- Key Concepts
- CoS in Your System
  - CoS Architecture
  - Configuring Priority Levels
  - Configuring a Rate Limit on Queue 1
  - Handling Tagged and Untagged Packets
- Standards, Protocols, and Related Reading

You can administer Class of Service (CoS) commands from the `bridge cos` *menu of the Administration Console. (See the* Switch 4007 Command Reference Guide.*) You can use the Administration Console after you log in to the Enterprise Management Engine and connect to a module slot.*

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

The Class of Service (CoS) options that are described in this chapter are available on Switch 4007 Layer 2 Switching Modules. For a similar but more extensive array of options on Multilayer Switching Modules, see the Quality of Service (QoS) information in Chapter 22.

Overview

Many network technologies, such as Ethernet and Fiber Distributed Data Interface (FDDI), have no inherent ability to distinguish between different types of traffic such as data, voice, and video, or even perhaps between different data applications. Thus, all traffic competes for the same bandwidth and is processed in a single queue by network devices. This approach to network service is described as "best effort" because there is no way to prioritize certain traffic ahead of other traffic.

If the network load is high and network devices become congested, certain bandwidth-intensive applications may receive a poor *quality of service (QoS)*. A jittery video conference display that does not reflect real-time movement or a crisp picture is an example of poor quality of service.

To overcome this limitation in Ethernet and FDDI, switch and router vendors developed a variety of QoS-oriented features that work at higher levels in the Open Systems Interconnection (OSI) model. Users can configure these features to better control how different types of traffic are processed and forwarded through the switch and ultimately the network as whole. QoS techniques are designed to address the different latency and throughput needs of time-sensitive applications, as well as to address the desire to prioritize business-critical information over non-critical information.

While QoS features clearly benefit a network with bandwidth constraints, they can add complexity and cost into network equipment and administration activities. Thus, the practical aim of the IEEE 802.1p standard is to outline a simplified version or subset of QoS techniques that preserves the high speed, low cost nature of traditional LAN bridging. Because the IEEE 802.1p standard addresses queuing and prioritization based on a numeric traffic class but it does not address bandwidth reservation or other approaches to QoS, the approach is often distinguished with the term *Class of Service (CoS)*.

Key Concepts

Before you configure CoS options in a Layer 2 Switching Module, review the following key concepts.

Basic Elements of the Standard

The two basic elements of the IEEE 802.1p standard are:

- Multiple processing queues in devices

  The standard does not require a specific number of queues but rather how different types of traffic could be allocated across up to eight queues.

- Priority levels carried in packets

  Packets that include priority levels are processed through the device queues in a way that is configured by the network administrator. The standard identifies eight different priority levels using numbers 0 through 7.

  Table 43 outlines the different types of traffic that the standards body envisioned carrying different priority levels. However, you can apply the eight numbers however you want to identify your network application traffic.

Table 43 Priority Levels and Traffic Types Envisioned by Standards Committee

Priority Level	Traffic Type
1	Background
2	(Spare)
0 (default)	Best Effort
3	Excellent Effort
4	Controlled Load
5	Video, less than 100 milliseconds latency and jitter
6	Voice, less than 10 milliseconds latency and jitter
7	Network Control

Format of Prioritized Packets

Priority level information can only be carried inside packets that are formatted according to the IEEE 802.1Q standard; such packets carry an extra 2 octets of data called a *tag*. The priority level information occupies 3 bits of this tag and VLAN information occupies 12 bits.

The following definitions summarize the difference between tagged and untagged packets and clarify two types of tagged packets:

- **Untagged packet** — Does not include an IEEE 802.1Q tag.
- **Tagged packet** — Includes an IEEE 802.1Q tag. There are two types:
  - **Priority-tagged packet** — Carries priority level information but no VLAN information.
  - **VLAN-tagged packet** — Carries priority level information and VLAN information.

Queues and Priority Levels

Compliance with the IEEE 802.1p standard means that a device must recognize eight priority levels (0–7), however the number of queues in a given device can vary. (Eight queues are not required.)

When there are fewer than eight device queues, a packet's priority level does not always indicate how it will be processed relative to other packets. This is because more than one priority level will be assigned to at least one of the queues. When multiple priority levels are assigned to the same queue, all packets in that queue are processed in the same manner, regardless of their priority level.

The characteristics of a given queue as well as overall product design determine how the packets in that queue are processed relative to packets in other queues. The device vendor identifies these characteristics.

CoS in Your System

Using the Administration Console on Layer 2 Switching Modules, you can:

- Enable or disable CoS (the setting affects all ports), which changes the number of hardware queues per port from one to two.
- Modify how the eight priority levels are assigned between the two queues.

  By default, the priority levels are assigned according to recommendations in the IEEE 802.1p standard. See "Configuring Priority Levels" later in this section.
- Set a rate limit on the high priority queue.

  See "Configuring a Rate Limit on Queue 1" later in this section.
- Display a summary CoS configuration.

CoS Architecture

When CoS is enabled, a Layer 2 Switching Module uses two CoS queues per port:

- Queue 1 is always the high priority queue.
  - Each Fast Ethernet port has a queue-specific buffer of 64 KB.
  - Each Gigabit Ethernet port has a queue-specific buffer of 128 KB.
  - You can affect the flow of queue 1 traffic by configuring a rate limit. See "Configuring a Rate Limit on Queue 1" later in this chapter.
- Queue 2 is always the low priority queue.
  - Each Fast Ethernet port has a queue-specific buffer of 256 KB.
  - Each Gigabit Ethernet port has queue-specific buffer of 512 KB.

When CoS is disabled, the high priority queues and associated buffers are shut off; all traffic flows through the low priority queues.

CoS settings are stored in non-volatile memory. Thus, in the event of a power cycle or reboot, user-configured settings are retained.

Important Considerations

- In non-blocking situations, CoS settings have no impact on traffic flow through the module.
- In blocking situations, queue 1 (high priority) traffic on a given port is processed ahead of queue 2 traffic on that same port. Traffic in queue 2 on that port is either delayed (buffered) or dropped (if buffers become full) as needed to allow the queue 1 traffic to be forwarded.
- Queue 1 traffic on a given port is not necessarily processed ahead of queue 2 traffic from other ports. This is because the switch selects traffic from each port in an approximate round-robin fashion.

Configuring Priority Levels

By default, CoS is enabled with priorities 4,5,6, and 7 assigned to queue 1 and priorities 0,1, 2, and 3 assigned to queue 2. This arrangement conforms with IEEE recommendations, but you can change it at any time.

Although there are two physical queues per port, the priority levels (traffic classes) that you assign to each queue actually apply to all ports in the module.

When you assign one or more priority levels to one of the queues, the module automatically assigns the remaining priority levels to the other queue.

If CoS is disabled, you can still modify the priority assignments to each queue; they simply do not effect traffic until you enable CoS.

Configuring a Rate Limit on Queue 1

You can configure a rate limit for queue 1. The rate limit is configured as a percentage of the number of packets received on each port.

The percentage refers to the number of packets that are processed from queue 1 out of every 8 packets received on the port. This *n of 8 packets* formula means that, in real terms, there are eight supported rate limit percentages: 12.5, 25, 37.5, 50, 62.5, 75, 87.5, and 100. The rate limit operates as a threshold, not as a bandwidth reservation technique.

Considering that the module does not accept decimal values, you must enter a whole number in the range that corresponds to one of the eight percentages. Depending on the number you enter, the module rounds down to the nearest *n of 8* value, although the summary display retains the number that you enter.

For example, if you enter any whole number between 88 and 99 as the rate limit, the working rate limit will be 87.5; that is, for every 8 packets received on a given port, 7 packets are selected from queue 1 and 1 packet is selected from queue 2.

Table 44 provides a reference chart:

Table 44 Implementation Guidelines for the CoS Rate Limit

Operating percentage	Range of values that you could enter	For every 8 packets received on the port and processed, the number of packets that are selected from queue 1
12.5	1–24	1
25	25–37	2
37.5	38–49	3
50	50–62	4
62.5	63–74	5
75	75–87	6
87.5	88–99	7
100	100	8

Important Considerations

- If the number of packets in queue 1 exceeds the rate limit, packets are held in the queue 1 buffer. When this buffer becomes full, the module begins to drop packets from that queue.
- The default rate limit of 100 percent means that queue 1 can "starve" queue 2 under the right conditions. That is, on a given port, packets in queue 2 are always buffered if there are any packets in queue 1 on that same port. Queue 2 packets are processed only after all packets from queue 1 have been processed. If the queue 2 buffers become full, the module begins to drop packets in that queue.

Handling Tagged and Untagged Packets

Consider the following points about how a Layer 2 module processes tagged and untagged packets with respect to CoS information:

- If CoS is enabled and an untagged packet enters a port, the packet is always processed through the low priority queue.
- As described earlier, the CoS priority level is carried in the IEEE 802.1Q tag. After the packet enters the module, its format is subject to change according to VLAN configurations and ingress and egress rules.
- If an untagged packet enters the module and the VLAN settings modify the packet to become a tagged packet, the module can insert VLAN information but cannot set a priority level other than 0 (whether CoS is enabled or not).
- If a tagged packet enters the module and the tag is retained upon forwarding the packet, the module leaves the priority level as-is (whether CoS is enabled or not).
- If a tagged packet enters the module and VLAN rules cause the tag to be stripped prior to forwarding, the CoS priority information is lost thereafter unless the packet is later processed by a device that can insert tags and priority levels other than 0.

Standards, Protocols, and Related Reading

The following standards provide more information about Class of Service:

- *IEEE 802.1p Traffic Class Expediting and Dynamic Multicast Filtering*

  A supplement to the IEEE 802.1D MAC Bridges base standard that addresses traffic prioritization in local area networks.

- *IEEE 802.1Q Virtual Bridged LANs*

  A base standard that specifies requirements for virtual LANs and a packet format that includes VLAN and CoS priority information.

To obtain copies of these standards, register for an on-line subscription with the Institute of Electrical and Electronics Engineers (IEEE) Web site:

`http://www.ieee.org`

11 IP MULTICAST FILTERING WITH IGMP

The Internet Group Management Protocol (IGMP) provides a way for a Switch 4007 Layer 2 Switching Modules to forward IP multicast application traffic to certain ports and filter it on other ports to increase bandwidth efficiency in the network. This chapter provides an overview, guidelines, and other key information about IGMP functions and their effect on IP multicast traffic. The chapter covers these topics:

You can manage IGMP commands on Layer 2 Switching Modules in either of these ways:

- *From the* `bridge multicast igmp` *menu of the Administration Console. (See the* Switch 4007 Command Reference Guide*).You can use the Administration Console after you log in to the Enterprise Management Engine and connect to a module slot.*
- *From an SNMP management application (not included with the system) using a private 3Com IGMP Snooping MIB, which is available from the 3Com Software Library on the Web.*

This chapter describes IGMP functions on Layer 2 Switching Modules. IGMP is also supported on Multilayer Switching Modules, but is described in the context of IP multicast routing. See Chapter 18 for more information.

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Overview

To transport their content to a community of network users, bandwidth-hungry applications often generate packets in the IP multicast format. Many network standards and protocols have been designed to create an efficient delivery system for IP multicast traffic from the source (usually a server) to the destinations (users). IGMP is one of these protocols.

If IGMP functions are disabled or not present in a Layer 2 switch, the switch floods all IP multicast packets to all ports — that is, it operates in compliance with the *IEEE 802.1D MAC Bridges* base standard. If IGMP functions are present and enabled, a switch can forward IP multicast traffic only to ports that require it and filter it on other ports.

Defined in Internet RFC 1112 and RFC 2236, IGMP performs two main functions in a Layer 2 switch: *snooping* and *querying*. Descriptions of these functions and how they work together are explained later in this chapter.

Benefits

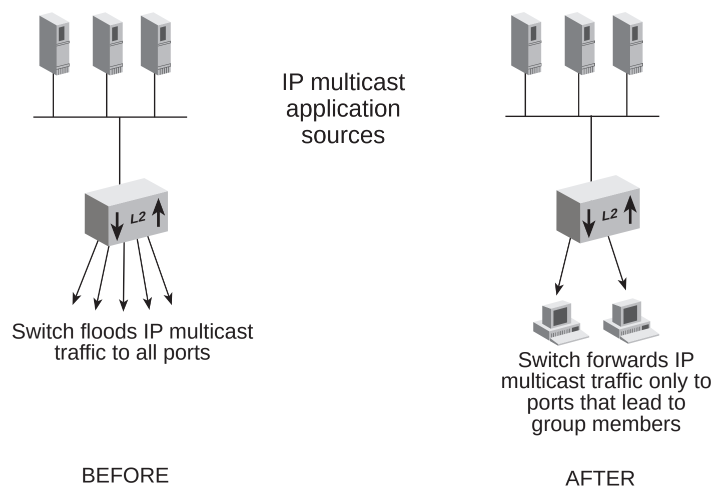
Support for IGMP in Layer 2 devices benefits your network in many ways:

- IGMP reduces the amount of bandwidth that an IP multicast stream would otherwise occupy at the edge of an IP multicast delivery tree; it is especially useful in flat network designs (large broadcast domains with cascading switches).
- IGMP reduces the amount of unwanted traffic that a host encounters on its Ethernet segment, which may be critical for users with low-speed connections to the network.

- IGMP requires minimal configuration in network devices and hosts. For example:
  - Snooping and querying functions can be easily enabled in switching devices.
  - IP-capable end stations do not usually require any special configuration because IGMP is already part of the IP protocol stack.
- Because more IP multicast applications are available each year, support for IGMP in switches helps prolong the life span of existing network topologies and available bandwidth.

To understand the fundamental benefit that IGMP provides for users attached to a switch, see Figure 8.

Figure 8 IP Multicast Traffic Flow Before and After IGMP Snooping



Key Concepts

IGMP plays a specific role in the overall delivery process for IP multicast traffic. Before you modify IGMP parameters in a Layer 2 Switching Module, review the following key concepts about IP multicast packets.

Devices That Generate IP Multicast Packets

Application sources (usually servers) generate IP multicast packets as the way to deliver their information (such as a video stream) to interested hosts (such as PC end stations).

Network devices generate IP multicast packets as the way to communicate with each other to establish a delivery path. These packets are issued by specific supporting protocols, such as IGMP.

Group Addresses and Group Members

An IP multicast packet differs from a unicast packet by the presence of a *multicast group address* in the destination address field of the IP header. Each application uses a unique group address, and hosts refer to these group addresses when they tell network devices which IP multicast transmissions they want to receive. In doing so, hosts become *group members*. Hosts can join and leave one or more groups at any time.

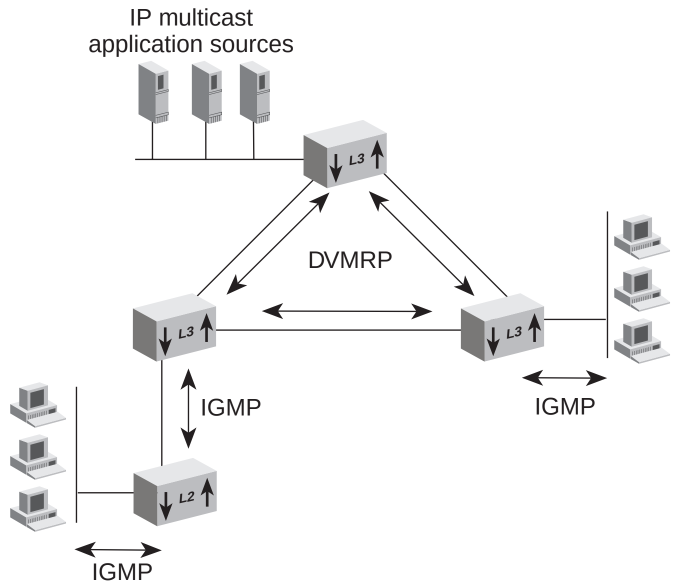
Communication Protocols

To coordinate an efficient, loopfree delivery path for IP multicast packets, certain protocols are used among network devices and hosts:

- A multicast routing protocol such as Distance-Vector Multicast Routing Protocol (DVMRP) is used between routers (if routers exist between sources and group members).
- A protocol such as IGMP is used between routers, switches, and hosts in each subnetwork or broadcast domain.

See Figure 9.

Figure 9 Protocols That Coordinate the Delivery of IP Multicast Traffic



Routers are not required for transmission of IP multicast packets between sources and group members. Compare Figure 8 and Figure 9; both represent valid designs in which IGMP can help conserve bandwidth.

IP Multicast Delivery Process

Even though there may be several, perhaps thousands of, intended recipients for a given IP multicast transmission, only one copy of a given packet is generated at the source. (This contrasts with a unicast approach, which would generate one copy per recipient.) The single copy of each IP multicast packet travels until the path to reach group members diverges, at which point the packet is replicated to ensure that one copy of the packet continues on each branch in the delivery tree. Thus, a significant benefit of the IP multicast delivery process is bandwidth conservation.

How Routers and Switches Use IGMP

Routers and switches use IGMP in similar ways:

- A router uses IGMP to determine which routing interfaces lead to group members.
- A switch uses IGMP to determine which port segments lead to group members.

Routers and switches both construct filters on ports that do not require group traffic to be forwarded. On each device, one group's traffic may be forwarded to one set of ports and another group's traffic may be forwarded to a different set of ports.

Tracking Group Member Locations

The ability to detect the location of group members is a product of two IGMP functions — querying and snooping — working together to construct individual delivery trees for each IP multicast group.

In each subnetwork or broadcast domain (VLAN), all switches and routers perform snooping (for their own connections), but only one of the devices needs to perform the querying. If there are multiple devices (routers and switches) in the subnetwork or broadcast domain that support querying, the one with the lowest IP address is elected as the *querier*.

The querier periodically sends a *query message* to all hosts on the subnetwork or broadcast domain and requests that they reply with the IP multicast groups for which they want to receive traffic.

A host responds to a query by sending an *IGMP report.* The querier as well as IGMP-capable devices between hosts and the querier *snoop* on the report's content to track which hosts (and their associated ports) belong to which groups.

If, after a certain period of time, a router or switch does not receive responses from any host on a given interface or port for a particular IP multicast group, the router or switch *prunes* the interface or port from the delivery tree to conserve network bandwidth.

If a switch that has downstream group members, an upstream router and upstream sources is elected as the querier, the switch must forward host reports to the upstream router to maintain the flow of IP multicast traffic to the switch.

How Hosts Use IGMP

Each host uses IGMP to communicate with the querier in a few different ways.

Host Membership Reports

Hosts transmit *Host Membership Reports (*hereafter called *IGMP reports)* in response to queries. A host sends a separate report for each group that it wants to join or to which it currently belongs. Hosts do not send reports if they are not or do not want to be group members.

Join Message

Rather than wait for a query, a host can also send an IGMP report on its own initiative to inform the querier that it wants to begin receiving a transmission for a specific group. This is called a *join* message. The benefit is faster transmission linkages.

Leave-Group Messages

Leave-group messages are a type of host message defined in IGMP version 2. If a host wants to leave an IP multicast group, it issues a leave-group message. Upon receiving such a message, the querier determines whether that host is the last group member on the subnetwork by issuing a *group-specific query*.

Leave-group messages lower *leave latency* — that is, the time between when the last group member on a given subnetwork or segment sends a report and when a router or switch stops forwarding traffic for that group. This process conserves bandwidth. The alternative is for the router or switch to wait for an aging period to expire before it ceases to forward the group traffic.

Report Suppression and Effect on Switch Activity

If host A hears an IGMP report from host B for the same group, host A suppresses (does not transmit) its own report for that group.

This approach was designed to conserve bandwidth, and it works well with routers because a routing interface only needs to know is that there is at least one group member in a subnetwork for it to forward group traffic onto that subnetwork.

However, because a switch that is not operating as the querier must forward IGMP reports to the querier, a switch must be able to track which ports lead to the querier, to routers, and to group members, so that it can forward IGMP reports only to the querier.

If the switch flooded IGMP reports, hosts on other segments would suppress their own reports for identical groups, which would cause the switch to set overly restrictive filters. Restricted forwarding of IGMP reports is necessary to allow the switch to receive IGMP reports from at least one host per group on each of its ports.

Configuring IGMP in Your System

Layer 2 Switching Modules support IGMP version 1 (RFC 1112) and version 2 (RFC 2236). You can manage the following IGMP options:

- Enable or disable the snooping function and the querying function.
  - Both settings apply to all ports on the module.
  - Both settings are disabled by default.
  - If enabled, the querying function requires that you configure a source IP address. See “Operating as the Querier” later in this chapter.
- Display a command configuration summary
- Display VLANs with active snooping activity
- Display group and port information per VLAN
- Display the designated querier per VLAN
- Display IP multicast router ports per VLAN
- Display the port in the VLAN that last received a query

With snooping and querying enabled, your switching module:

- Builds for each VLAN a table of member ports per IP multicast group and adjusts the table over time through an aging mechanism as well as by processing IGMP messages.
- Sets per-port filters that allow targeted forwarding of IP multicast group traffic.
- Determines which ports lead to routers. See “Locating Multicast Routers” in this chapter.
- Determines which port leads to the querier (unless the module is the querier). See “Operating as the Querier” in this chapter.
- Supports a private IGMP Snooping MIB (`3cigmpSnoop.mib`).

Key Implementation Guidelines

Consider these points when you configure IGMP options in a Layer 2 Switching Module:

- IGMP snooping and querying works for *IP* multicast packets only. Other protocol-based multicast packets are flooded to all ports in compliance with the IEEE 8021.D standard.
- Each IGMP-capable device performs snooping for its own ports, but only one device in a subnetwork or broadcast domain acts as the querier for all devices. Thus, on a single switching module, you can enable snooping and disable querying as long as one or more other devices in the subnetwork or broadcast domain can act as the querier.
- If you enable querying on a given module, you must also enable snooping for querying to work properly.
- To maximize the effectiveness of IGMP in a flat network design or large broadcast domain that includes IP multicast sources, the querier should be positioned as close to the source of IP multicast traffic (usually servers) as possible. You can accomplish this by enabling the query function only on certain devices.
- Because some IP multicast applications transmit a large number of unsolicited packets or may require security protection, 3Com recommends that you place IP multicast sources upstream from a Layer 3 switch or router.

- If you have configured Open VLAN mode and IP multicast packets are tagged (IEEE 802.1Q format), then the IGMP tables in each VLAN share information with each other. VLANs do not form barriers in the flow of IP multicast traffic, even though they form separate broadcast domains. IGMP operates as if there was a single broadcast domain. However, if you have configured Open VLAN mode and the packets are not tagged, then VLANs do form barriers and the IP multicast traffic is restricted.
- If you have configured Closed VLAN mode, then the IGMP tables do not share information and forwarding of IP multicast packets is restricted to the ports in the VLAN on which the IP multicast packet arrives. If all VLANs contain potential IP multicast group members, then you would need to have a router connection to each VLAN to ensure IP multicast packet delivery to those group members.
- Ensure that any rate limit that you implement with the `bridge port multicastLimit` command does not interfere with the flow of IP multicast traffic. If you select the `BcastOnly` option, the rate limit does not affect multicast packets. For more information about the bridge port multicast limit feature, see Chapter 9.
- To reduce the effects of MAC address aliasing, verify that your IP multicast applications do not use binary group addresses in the range [224 – 239]. [0,128].0.x, where x equals 0 – 255. See "Effects of MAC Address Aliasing" later in this chapter for more information.
- Your switching module supports both IGMP version 1 and 2. For maximum benefit, verify that the IP stack in your host endstations also supports both IGMP version 1 and 2.
- If a resilient link pair exists between two switches that are downstream from the IP multicast source and IGMP querier and the main link in the pair fails over to its standby link, the transmission to group members is temporarily interrupted. The IP multicast transmission resumes after hosts send IGMP join messages or after the next query travels down the [now active] standby link and hosts respond with IGMP reports. Queries are sent every 125 seconds. Depending on when the last query was sent prior to the link failover, the interruption can last up to approximately 2 minutes.

Processing IP Multicast Packets

Table 45 summaries how a Layer 2 Switching Module processes various types of IGMP packets and other IP multicast packets.

Table 45 How the System Processes IP Multicast Packets

Packet Type	Is Forwarded To*
IGMP Membership Reports	Ports in the broadcast domain that lead to multicast routers Port that leads to the querier (if another device is the querier)
IGMP Queries	All ports in the broadcast domain
IGMP Leave-Group Messages	All ports in the broadcast domain
Packets with addresses [224 – 139].[0,128].0.x where x = 0 – 255	All ports in the broadcast domain (See "Effects of MAC Address Aliasing" later in this chapter.)
Packets addressed to known (registered) IP multicast group	Ports on which IGMP membership reports for that group have been heard Ports that lead to multicast routers Port that leads to the querier (if another device is the querier)
Packets addressed to unknown IP multicast group (group for which no host has registered)	No ports, except those that lead to multicast routers and the querier (if another device is the querier).

* Except for the port on which the packet originated.

Some ports may not be available for carrying traffic. Two examples are: ports that have been administratively disabled or ports that the Spanning Tree Protocol has prevented from being in the forwarding state.

Effects of MAC Address Aliasing

Operating as a Layer 2 device, your module filters IP multicast traffic by referring to hexidecimal MAC addresses that correspond to binary IP multicast group addresses.

A multicast MAC address is created by selecting only the low order 23-bits in the Class D binary IP address, translating that portion to hexidecimal format, and attaching it to a standard set of bits that signify it is a multicast packet (01-00-5E). For example, IP address 224.10.8.5 becomes MAC address 01-00-5E-0A-08-05.

Because only a portion of the binary IP address is translated, several different binary addresses can map to the same hexidecimal MAC address. This situation is called *MAC address aliasing.*

Because a switching module cannot distinguish such packets, MAC address aliasing has two main implications:

- Some packets are forwarded to more ports than actually require it.

  For example, if requests for multicast group 226.1.2.3 are registered on port 1 and requests for group 227.1.2.3 are registered on port 2, these IP addresses map to the same MAC address and the module forwards traffic for both groups to both ports.

- Packets with certain addresses can never be filtered by IGMP.

  In most cases, such packets are routing protocol advertisements that use addresses from the block of permanent reserved addresses that is administered by the Internet Assigned Numbers Authority (IANA). To ensure these advertisements make their way through the network, it is important that these packets do not get filtered by IGMP. However, if an IP multicast application uses a group address that maps (due to MAC address aliasing) to one of these permanent addresses, these packets cannot be filtered by IGMP either.

Important Considerations

- To reduce the effects of MAC address aliasing, verify that your IP multicast applications do not use binary group addresses in the range [224 – 239]. [0,128].0.x, where x equals 0 – 255.
- See Table 46 for several examples of permanent reserved addresses. For a complete and current list, visit the Web site of the Internet Assigned Numbers Authority (IANA) at:

  `http://www.iana.org`

Table 46 Examples of Class D Permanent Address Assignments

Address	Meaning
224.0.0.0	Base Address (Reserved)
224.0.0.1	All systems on this subnet
224.0.0.2	All routers on this subnet
224.0.0.4	All DVMRP routers
224.0.0.5	All OSPF routers
224.0.0.6	All OSPF designated routers
224.0.0.7	All ST routers
224.0.0.8	All ST hosts
224.0.0.9	All RIP version 2 routers
224.0.0.11	Mobile agents
224.0.0.12	DHCP server/relay agent
224.0.0.13	All PIM routers
224.0.0.14	RSVP, Encapsulation
224.0.0.15	All CBT routers

Operating as the Querier

For a Layer 2 Switching Module to offer itself as a potential IGMP querier for its subnetwork or broadcast domains (VLANs), you must:

1. Enable the IGMP snooping option.

   The module cannot send queries if snooping is disabled.

2. Enable the IGMP querying option.
3. Configure an IP address that the module can insert as the source address of query packets.

   Use the `bridge multicast igmp queryIpAddress` command.

As the querier, the module sends general queries every 125 seconds and group-specific queries when prompted by host leave-group messages.

Locating Multicast Routers

A switching module must be able to identify which of its ports are connected to multicast routers so that it can forward appropriate IP multicast traffic to them. For example:

- If the module is operating as the querier, any upstream router (the router that leads back toward an IP multicast source) also needs to see IGMP reports so it can decide whether to begin, stop, or continue forwarding group traffic on the subnetwork that includes the switch.
- Downstream routers need to receive all IP multicast traffic because other group members may be attached to them.

A switching module records which of its ports lead to IP multicast routers by snooping on protocol advertisements that are sent by the routers. The module can recognize the advertisements of the following multicast routing protocols:

- Distance Vector Multicast Routing Protocol (DVMRP)
- Multicast Open Shortest Path First (MOSPF)
- Protocol Independent Multicast (PIM) version 1
- Protocol Independent Multicast (PIM) version 2

The module uses a 100-second interval to age the records it keeps for multicast router locations. Protocol advertisements are sent much more frequently than this.

Aging the IGMP Tables

If a switching module receives no host reports for a given group on a given port within a certain period of time (the aging interval), it ages that entry in its IGMP tables and sets a filter for that group on that port.

The aging interval is the period from which the last query is sent until the time when the module stops forwarding traffic on that port. The aging interval is determined in the following way: multiply the query interval (125 seconds) by two and then add one query response interval (10 seconds). Thus, it is approximately 4.5 minutes.

Standards, Protocols, and Related Reading

The following standards apply to the technologies that are described in this chapter:

- *IEEE 802.1D Media Access Control (MAC) Bridges*

  A base standard that specifies requirements for transparent bridging. To obtain copies of standards, register for an on-line subscription the Institute of Electrical and Electronics Engineers (IEEE) Web site:

  `http://www.ieee.org`

- RFC 1112: *Host Extensions for IP Multicasting*. S. Deering, July 1986.
- RFC 2236: *IGMP version 2*. W. Fenner, November 1997.

  To obtain copies of Internet RFCs and proposed standards, visit the Internet Engineering Task Force (IETF) Web site:

  `http://www.ietf.org`

12 TRUNKING

This chapter provides guidelines, limitations, and other important information about how to implement the trunking function on Switch 4007 modules.

This chapter covers the following topics:

- Trunking Overview
- Key Concepts
- Key Guidelines for Implementation
- Automatic Backplane Trunking
- Defining Trunks
- Modifying Trunks
- Removing Trunks
- Standards, Protocols, and Related Reading

You can manage trunks in either of these ways:

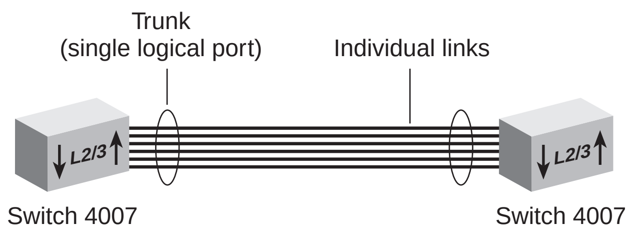
- *From the* `bridge trunk` *menu of the Administration Console. (See the* Command Reference Guide.*) You can use the Administration Console after you log in to the Enterprise Management Engine and connect to a module slot in the Switch 4007 chassis.*
- *From the Bridge trunk folder of the Web Management software. (See the* Switch 4007 Getting Started Guide.*)*

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Trunking Overview

A *trunk* (also known as an *aggregated link*) works at Layer 2 and Layer 3 of the Open Systems Interconnection (OSI) model and allows you to combine multiple Fast Ethernet and Gigabit Ethernet ports on interface modules into a single high-speed link between two switches. See Figure 10.

Figure 10 Conceptual Example of a Trunk



An interface module treats trunked ports in the same way that it treats individual ports. Also, all higher-level network functions — including Spanning Tree algorithms, virtual LANs (VLANs), and Simple Network Management Protocol (SNMP) management — do not distinguish a trunk from any other network port.

Features

You can configure the following trunking features:

- **Define** — You specify ports and characteristics associated with the trunk.
- **Modify** — You modify a trunk's characteristics or add or remove a port from the trunk.
- **Remove** — You remove a trunk definition from the module.

Benefits

Trunking can help you meet your network capacity and availability needs. With trunks, you can cost-effectively increase the bandwidth between switches or between servers and switches as your network requires. With trunking, you combine multiple Fast Ethernet and Gigabit Ethernet ports into a single high-speed channel.

If Gigabit Ethernet is not available, you can use trunked Fast Ethernet to increase network capacity. After Gigabit Ethernet is in place and the time comes to scale links beyond 1000 Mbps, you can use trunking to create multigigabit connections.

Trunks also enhance network availability because the Trunk Control Message Protocol (TCMP) detects and handles physical configuration errors in the point-to-point configuration. The interface module automatically distributes traffic across the ports that are associated with the trunk. If any of the trunk's ports go down or up, the module automatically redistributes traffic across the new arrangement of

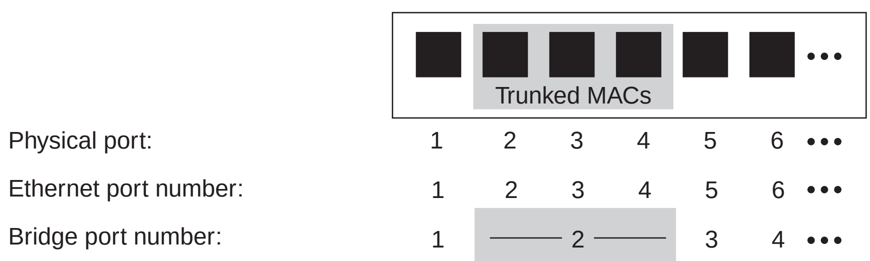
Key Concepts

Before you configure trunking on your module, become familiar with the key concepts in this section.

Port Numbering in a Trunk

When you combine ports on a trunk, the module logically groups the physical ports that you specify into a single bridge port, identified by a single bridge port number in bridge statistics. For example, Figure 11 shows that Ethernet ports 2, 3, and 4 are represented by bridge port 2 after trunking. The lowest numbered port in the trunk, called the *anchor port*, represents the entire trunk. After trunking, you can select bridge port 2 when you specify bridge port or virtual LAN (VLAN) information, but you cannot select bridge ports 3 or 4 since they are part of the trunk.

Figure 11 Bridge Port Numbering After Trunking



Regardless of whether you define trunking, the physical port numbering on your module remains the same.

It is important to understand the relationships between Ethernet, bridge, and VLAN port-related information:

- **Ethernet port information** — Each physical port is always listed individually, regardless of whether it is part of a trunk.
- **Bridge port information** — This information uses the concept of bridge ports. When you perform bridge port operations, you specify the trunk's anchor port, not the other ports in the trunk, as the representative bridge port. In the bridge port displays, each selectable bridge port has a port field that contains multiple port numbers if the bridge port represents a trunk (for example, `3,5` or `6-8`).
- **VLAN information** — When you define VLANs (as described in Chapter 14), you must specify the bridge ports that you want to be part of the VLAN. If you have a trunk, you specify its anchor port as the bridge port. The VLAN that you create then includes all of the physical ports in the trunk.

  If you plan to use trunks (aggregated links), define the appropriate trunks *before* you define your VLANs. (If you define a VLAN with certain ports and subsequently configure some of those ports to be part of a trunk, the module removes those ports from the VLAN and places them in the default VLAN.) When you define a VLAN that includes trunk ports, you must specify the trunk's anchor port (lowest-numbered port).

Trunk Control Message Protocol (TCMP)

The Trunk Control Message Protocol (TCMP) performs the following functions:

- Detects and corrects trunks that violate trunk configuration rules
- Ensures orderly activation and deactivation of trunk ports

The module runs a separate TCMP agent for each trunk. If TCMP detects an invalid configuration, the protocol restricts the trunk to the largest subset of ports that is a valid configuration.

By default, TCMP is enabled. Keeping TCMP enabled is optional, but recommended. If you disable TCMP, the network still functions, but without automatic trunk validation and reconfiguration. By default, TCMP is enabled.

Each TCMP agent:

- Periodically transmits a TCMP helloMessage through every trunk port.
- Continuously listens for helloMessages from other trunk ports.
- Builds a list of ports that TCMP has detected.
- Uses this list to activate or deactivate trunk ports to maintain valid trunk configurations.

TCMP uses three trunk port states to control port activation and deactivation:

- **notInUse** — A trunk port in this state has not been *selected* to participate in the trunk.
- **selected** — TCMP has *selected* the trunk port to participate in the trunk, but the port has not yet become *active*.
- **inUse** — A trunk port is fully *active* on the trunk.

Key Guidelines for Implementation

Consider the following important factors when you implement and configure trunks:

General Guidelines

- Create trunks before you define VLANs.
- An interface module supports four point-to-point trunks, each built from up to eight ports. All channels in a trunk must connect:
  - Correctly configured ports
  - Identical types of ports (with no two ports on a trunk connected to the same network)
  - Identical types of network nodes (switches or servers)
- You cannot mix Fast Ethernet and Gigabit Ethernet links in a trunk. All links to be trunked must be homogeneous.
- You can create a trunk on modules through the backplane using a switch fabric module and any other switching module.

- When multiple links are trunked, it can be difficult to manage and troubleshoot individual port-to-port connections if a connectivity problem occurs. This issue may not be of concern in a server farm room. But if you use trunking extensively between wiring closets and data centers, the large number of connections involved and their distributed nature may make their management and troubleshooting difficult. 3Com recommends that you apply trunking only *within* data center and campus interconnect areas.

  3Com uses a three-tiered framework to describe the different functional areas in a local area network (LAN):

  - **Wiring closet** — This area provides connections to user workstations. It also includes downlinks into the data center or campus interconnect.
  - **Data center** — This area receives connections from wiring closets and campus interconnect areas. Most local server farms reside here.
  - **Campus Interconnect** — This area only appears as a separate location in larger networks; smaller networks usually have just wiring closets and data centers. The campus interconnect links campus data centers to each other, and may also include an enterprise server farm and connections to a wide area network.

- 3Com recommends that you use trunks to increase network availability in the following scenarios:

  - Switch-to-switch connections in the data center and campus interconnect areas
  - Switch-to-server connections in the data center and campus interconnect areas
  - Downlinks from the data center to the campus interconnect

- The trunking feature in 3Com switches is currently a proprietary implementation. No *de facto* standards exist.

Trunk Capacity Guidelines

- The device-to-device burst-transmission rate across a trunk is limited to the speed of just *one* of the port-to-port links within the trunk. For example, the maximum burst rate over a 400-Mbps pipeline with four trunked Fast Ethernet links is 100 Mbps. This limitation preserves frame ordering between devices, usually by moving all traffic between two specific MAC addresses across *only one port-to-port link*. Therefore, trunking provides no direct benefit for some one-way applications, such as server-to-server backups. This limit exists for most vendor implementations.
- The total throughput of a trunk is typically less than the bandwidth obtained by adding the theoretical capacity of its individual links. For example, four 1000-Mbps links do not yield a 4000-Mbps trunk. This is true with all vendor implementations.
- A trunked Fast Ethernet pipeline may seem to offer comparable bandwidth to a single Gigabit Ethernet link, and trunked Fast Ethernet may seem like a good way to buy some time before you upgrade connections to Gigabit Ethernet. Table 47 shows that trunking Fast Ethernet may not be an effective strategy.

  If you cannot upgrade to Gigabit Ethernet, then trunking Fast Ethernet in switch-to-switch or switch-to-server links can help you fine-tune or expand network capacity. After Gigabit Ethernet is in place, you can use trunking to further expand switch-to-switch or server-to-switch links.

Table 47 Comparing Gigabit Ethernet with Trunked Fast Ethernet

Comparison Point	Gigabit Ethernet	Trunked Fast Ethernet
Max burst rate	1000 Mbps	100 Mbps
Max aggregate rate	1000 Mbps (2000 Mbps full duplex)	Multilayer modules: 800 Mbps (over 8 links) (1600 Mbps full duplex) Layer 2 modules: 600 Mbps (over 6 links) (1200 Mbps full duplex)
Standards compliance	IEEE 802.3z	In progress

Automatic Backplane Trunking

Provides automatic backplane trunking on both the switch fabric module and the managed interface modules (such as the 4-port GBIC GEN Interface Module). You can enable or disable this feature at the switch fabric module for any or all of the interface modules.

The 24-port Gigabit Switch Fabric Module (Model Number 3CB9FG24T) utilizes the chassis backplane traces on a slot-by-slot basis. Trunking is performed through the switch fabric module backplane and not through the interface modules.

Important Considerations

- Automatic backplane trunking is provided only through the switch fabric module.
- If automatic backplane trunking is enabled, then you cannot configure backplane port trunking on the switch fabric module or interface modules. If you enable a slot, that slot automatically trunks any backplane ports.
- If automatic backplane trunking is disabled, then you can configure backplane trunking on the switch fabric module and have those trunks automatically configured at the managed interface module. This centralizes the trunking application through the switch fabric module.
- Staging is not supported if autoMap is enabled on the module.
- The autoMap feature does not support single-port trunk groups. No backplane trunks containing fewer than two ports can be defined on the switch fabric module. However, on the interface module, a single port trunk is allowed only when the switch fabric module has a trunk already defined.

- When automatic backplane trunking is *enabled*, consider these issues:
  - No trunking can be performed on the 2-port or 4-port GEN interface module.
  - Layer 2 and Multilayer Switching Modules with multiple backplane ports can use the maximum trunk groups available from the switch fabric module to the specific slot.
  - Layer 2 and Multilayer Switching Modules with a single backplane port cannot be trunked.
- When automatic backplane trunking is *disabled*, consider these issues:
  - Trunking can be performed on the 4-port GEN interface module between slots containing other GEN interface modules.
  - Trunking is allowed on a minimum of two backplane port when using Multilayer Switching Modules with multiple backplane ports
  - Trunking is allowed on Layer 2 modules with trunk max traces available
  - No trunking is allowed on either Layer 2 or Multilayer Switching Modules with a single backplane port

Defining Trunks

To define a trunk, you specify the ports that you want to be in the trunk.

Important Considerations

- If you have already defined other trunks on your Switch, you cannot select ports that are part of an existing trunk.
- Devices that you use in a trunking configuration must have the hardware to support the trunking algorithm.
- You can define one or several trunks using a single `define` command. This capability saves you from having to reboot the Switch after each trunk define.
- When you define a trunk, you specify ports and characteristics associated with the trunk (including Gigabit Ethernet flow control). You can specify them all in one `define` operation.
- When you create the trunk, the entire trunk assumes the current port characteristics.
- Trunk names cannot be longer than 32 characters (no spaces allowed).

- 3Com recommends that the TCMP state be `enabled`. But devices can operate without TCMP. When TCMP is not in effect on a point-to-point link, its configuration validation is simply absent.
- If your Switch has more than one media type (for example, Fast Ethernet and Gigabit Ethernet), you are prompted for a media type before you are prompted for the trunk information.
- Trunk names become the port labels when you display information on the trunks.
- All ports in the trunk are set to the selected operating mode (half-duplex or full-duplex).
- Each Gigabit Ethernet (GEN) Switch Fabric Module (Model Number 3CB9FG24T) that you install provides port trunking support for 12 groups, with up to six ports in a group (12x6).
- When you create a VLAN that includes ports that are part of a trunk, specify the *anchor* port (lowest-numbered port) that is associated with the trunk. For example, if ports 1 through 3 are associated with a trunk, specifying port 1 defines the VLAN to include all of the physical ports in the trunk. If you have not defined trunks, simply specify one or more port numbers, or specify `all` to assign all ports to the VLAN interface.
- When you create a trunk that includes ports that are part of a VLAN, those ports are removed from the VLAN. You must modify the VLAN and add the new bridge port to the appropriate VLAN. This situation does not apply to the default VLAN (all ports are part of the default VLAN, including the trunk's anchor port).
- Layer 2 modules have two backplane ports and Multilayer modules have one backplane port. If you have more than one backplane port between the fabric module and the interface module, you can trunk those ports together.

  You have to setup trunking on both the interface module and the switch fabric module sides. After you do this, the trunk comes up, but only one port is active within the trunk. Therefore, before you define the trunks using backplane ports, manually enable the second backplane port on the interface module.
- Entering an `nvData reset` command erases all previous trunk information.

Modifying Trunks

You can modify a trunk in two ways:

- You can modify a trunk's characteristics (for example, the operating mode or the TCMP state).
- You can add or remove a port from the trunk.

Important Considerations

- You must keep at least one port that you defined in the original trunk. To completely redefine a trunk configuration, remove the trunk and define a new one.
- You cannot modify, add, or remove ports that are part of different trunks from the one you that you are modifying.
- To avoid configuration errors, do not modify FDDI station mode port-pairs when any of the ports in the pair are members of a trunk.
- If you have more than one media type on your Switch (for example, Fast Ethernet and Gigabit Ethernet), you are prompted for a media type before you are prompted for the trunk information.
- Any changes that you make to the trunk's characteristics take effect immediately and do not interrupt trunk operations. If you add or remove a port, however, you must reboot the Switch to implement the change.
- You can modify one or several trunks using a single `modify` command.

Removing Trunks

You can remove one, several, or all trunks using a single `remove` command. This capability saves you from having to reboot the Switch after each trunk remove.

Important Consideration

- If you remove a Gigabit Ethernet module that has trunks defined, NVRAM is not cleaned up, but the trunk ports are available for use by a replacement module of the same type.

Standards, Protocols, and Related Reading

The Switch 4007 supports these Ethernet standards:

- **IEEE 802.3** — 10BASE-T Ethernet over unshielded twisted pair (UTP)
- **IEEE 802.3u** — 100BASE-T Fast Ethernet over UTP or fiber
- **IEEE 802.3z** — 1000BASE-SX Gigabit Ethernet over multimode fiber and 1000BASE-LX Gigabit Ethernet over multi- or singlemode fiber

Although the standard for trunking (link aggregation) is not yet finalized, 3Com trunking technology currently interoperates with similar technology from other vendors, including Sun Microsystems and Cisco Systems.

13 RESILIENT LINKS

This chapter provides an overview, guidelines, and other important information about how to implement resilient links on Layer 2 Switching Modules in your Switch 4007 system.

The chapter covers these topics:

- Resilient Links Overview
- Key Concepts
- Key Guidelines for Implementation
- Resilient Link Define and Modify
- Resilient Link State
- Resilient Link Active Port
- Resilient Link Remove

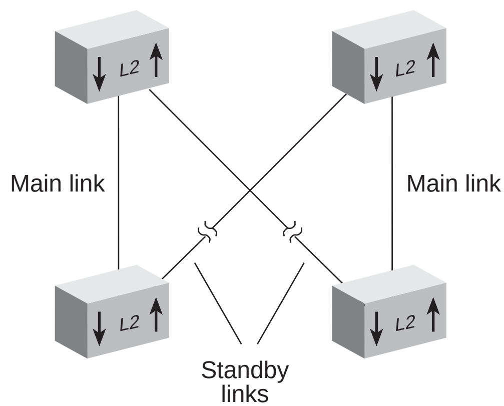
After you log in to the Management Module (EME) and connect to a slot that houses a Layer 2 Switching Module, you can manage resilient links from the `bridge link` *menu of the Administration Console. For more information on specific commands, see the* Switch 4007 Command Reference Guide.

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Resilient Links Overview

Resilient links protect your network against an individual link or device failure by providing a secondary backup link that is inactive until it is needed. A resilient link comprises a resilient link pair that contains a main link and a standby link. If the main link fails, the standby link immediately takes over the task of the main link. Figure 12 shows a resilient link pair.

Figure 12 Resilient Link Pair



Under normal network conditions, the main link carries your network traffic. If a signal loss is detected, the device immediately enables the standby link so that it carries the data and sends a trap to the network management station to alert you of the signal loss. The standby port assumes the profile and carries the network traffic of the main port.

If the main link has a higher bandwidth than its standby link, traffic is switched back to the main link, provided that no loss of link is detected for 2 minutes. Otherwise, you must manually switch traffic back to the main link.

Switchover time to the backup link takes less than 1 second, ensuring no session timeouts and therefore seamless operation.

To keep you informed about network activity, configure the module to generate an SNMP trap whenever a switchover from one link to the other occurs or whenever the link state (up or down) of either link in the resilient pair changes. For more information about how to configure and enable SNMP traps, see Chapter 23.

Features You can configure these features for resilient links:

- **Define** — Specify a name and the ports that you want to associate with the link.
- **Link state** — Enable or disable a resilient link pair.
- **Active port** — Define either port as the port that carries network traffic.
- **Modify** — Change the name and ports that are associated with a previously defined resilient link.
- **Remove** — Delete one or more resilient links from the module.

Benefits

- Resilient links enable you to protect critical links and prevent network downtime if those links fail.
- 3Com recommends that you implement resilient links in these network configurations:
  - Switch-to-switch downlinks from the wiring closet to the data center. The resilient link pair must terminate on a Layer 2 data-center switch.
  - Server-to-switch connections in the data center and campus interconnect areas.

Key Concepts

- When you define a resilient link pair, you define:
  - **The main port** — The port on which network traffic runs under normal operation.
  - **The standby port** — The port to which network traffic shifts if the main port fails.
- You can define either the main port or the standby port as the *active* port, that is, the port that is carrying network traffic. The main port is usually defined as the active port.

Key Guidelines for Implementation

Consider these important factors when you implement and configure resilient links.

General Guidelines

- Create resilient links before you define your VLANs. If you plan to create resilient links to be part of a VLAN, create the resilient links before you create the VLAN.
- You must reboot the module when you finish defining the resilient links. (You can define multiple links in one `define` operation.)
- If you perform an `nvData reset` operation, the module erases all previous resilient link settings.
- Resilient links are a point-to-point technology; they do not react to "downstream" network failures.
- Inactive links take up ports but do not add to network capacity.
- You cannot set up resilient link pairs if the Spanning Tree Protocol (STP) is enabled.
- You cannot disable ports that are part of a resilient link unless a link failure occurs.
- You need to define a resilient link only at one end of the link.
- If an active standby port fails and you have defined a link on the main port, the ports toggle and the main port becomes active.

Resilient Link Define and Modify

To define or modify a resilient link, specify the ports that you want to be in the resilient link.

Important Considerations

- When you define or modify a resilient link, you specify the name of the link and the ports that you want to associate with the link. You can specify all ports in one `define` operation.
- You can define more than one resilient link at a time, so that you reboot the module only once.
- When you create a resilient link that includes ports that are part of a VLAN, those ports are removed from the VLAN. You must modify the VLAN to add the new bridge port. This consideration does not apply to the Default VLAN. All ports are part of the Default VLAN.

- If you have already defined other resilient links on your module, you cannot select ports that are part of an existing resilient link to be part of an additional resilient link pair.
- You cannot select a trunked port nor the trunk itself as part of a resilient link.
- The resilient link name can be up to 32 characters long.

See the *Switch 4007 Command Reference Guide* for a complete description of the resilient link commands.

Resilient Link State

You can enable or disable one or more resilient link pairs with a single command.

Important Considerations

- When the link state is *enabled*, the resilient link can transmit and receive traffic.
- When the link state is *disabled*, the resilient link no longer transmits or receives frames.

Resilient Link Active Port

The active port is the port that carries traffic. You can designate either the main port or the standby port as the active port.

Important Considerations

- Only one port in a resilient link pair is active at a time.
- By default, the module defines the main port in a resilient pair as the active port when you reboot, unless the main link is down.
- If the main link is of equal or lesser bandwidth than the active standby link, the switchover back to main link is not automatic. If you want the main link to be active, you must configure it as the active port.

Resilient Link Remove

You can remove one or more resilient links with a single remove command.

Important Consideration

- After you remove a resilient link pair, you must reboot the module.

14 VIRTUAL LANs (VLANs)

This chapter provides guidelines and other key information about how to manage VLANs on your Switch 4007.

This feature is available on Layer 2 and Multilayer Switching Modules. Differences in implementation between these two module groups are noted where applicable.

The chapter covers these topics:

- VLAN Overview
- Key Concepts
- Key Guidelines for Implementation
- VLAN allOpen or allClosed Mode
- Port-based VLANs
  - The Default VLAN
  - User-Configured Port-based VLANs
  - Dynamic Port-based VLANs Using GVRP
- Protocol-based VLANs
- Network-based IP VLANs
- Ignore STP Mode
- Rules of VLAN Operation
- Modifying and Removing VLANs
- Monitoring VLAN Statistics

The management interfaces display "`cb9000`" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.

You can manage VLAN features in either of these ways:

- *From the* `bridge vlan` *menu of the Administration Console. (See the* Command Reference Guide*.) You can use the Administration Console after you log in to the system and connect to a module slot.*
- *From the Bridge VLAN folder of the Web Management software. (See the* Switch 4007 Getting Started Guide*.)*

VLAN Overview

A *virtual LAN (VLAN)* is a logical grouping that allows end users to communicate as if they were physically connected to a single LAN, independent of the physical configuration of the network. A VLAN is generally considered equivalent to a Layer 2 broadcast domain or a Layer 3 network.

Your system's point of attachment to a given VLAN is called a *VLAN interface*. For the Switch 4007, a VLAN interface exists entirely within a single switching module or switch fabric module; you control the configuration of the VLAN interfaces. A VLAN and a VLAN interface are analogous to an IP subnetwork and an IP interface on a router.

Need for VLANs

If a bridge port in a LAN switching device receives a frame with a broadcast, multicast, or unknown destination address, it forwards the data to all bridge ports in the VLAN that are associated with the frame, except the port on which it was received. This process is referred to as bridge *flooding*. As networks grow and the amount and types of traffic increase, bridge flooding may create unnecessary traffic problems that can clog the LAN.

To help control the flow of traffic through a switching device and meet the demands of growing networks, vendors have responded by:

- Using customized packet filtering and IP multicast controls such as Internet Group Management Protocol (IGMP) snooping to provide further controls on which packets are forwarded through the bridge. These controls require more complex configuration by the administrator.
- Using more and more routers as broadcast firewalls to divide the network into broadcast domains. As the number of legacy routers increase, *latency* begins to degrade network performance, increase administration overhead, and raise operating costs.

- Using the Spanning Tree algorithm in switching devices to control the flow of traffic among LANs (for redundant links). These mechanisms work best only in certain types of LAN topologies.

VLAN technology provides a high-performance and easy-to-implement alternative to routers for broadcast containment. When you use switching devices with VLANs:

- Each network segment can contain as few as one user (approaching private-port LAN switching), while broadcast domains can be as large as 1,000 users, or even more.
- VLANs can help network administrators track workstation movements to new locations without manual reconfiguration of IP addresses.
- VLANs can be used to isolate unicast traffic to a single broadcast domain, thereby providing a form of network security.

Benefits

You can use VLANs to:

- Reduce the cost of equipment moves, upgrades, and other changes and simplify network administration.
- Create virtual workgroups in which members of the same department or section appear to share the same LAN, with most of the network traffic staying in the same VLAN broadcast domain. They can isolate broadcast and multicast traffic to a single broadcast domain, as well as unicast traffic.
- Help avoid flooding and minimize broadcast and multicast traffic.
- Reduce the need for routing to achieve higher network performance, ease of administration, and reduced costs.
- Control communication among broadcast domains.

VLANs on the Switch 4007

Your system offers a collection of modules that pass traffic to one another using a central switch called the Gigabit Ethernet (GEN) Switch Fabric Module. This switch fabric module, operating at Layer 2, controls the Ethernet traffic associated with its modules.

The switch fabric module supports a variety of Layer 2 Switching Modules, Multilayer Switching Modules, and Interface Modules. See the Switch 4007 Getting Started Guide for a list of supported modules.

The examples in this chapter represent the location of the switch fabric module logically to emphasize its central role in the configuration process.

To create VLANs in the Switch 4007 environment, you configure these components:

- **Layer 2 and Multilayer Switching Modules** — You connect to these individually through the EME (Enterprise Management Engine) and configure the following ports based on your VLAN configuration:
  - **Front-panel ports** — Typically, these ports connect external devices to the switching module. The number of front-panel ports varies according to the model of switching module.
  - **Backplane port** — Connects the switching module to the switch fabric module.
  - Example: If you are logged into a 20-port Layer 2 Switching Module and configure front-panel ports as members of a port-based VLAN that spans modules, you configure port 21 (the lowered-numbered backplane port) as part of the VLAN. When you have multiple VLANs, this module backplane port must be tagged for all but one of the VLANs. (For one VLAN, such as the default VLAN, the backplane port can be untagged, but for the other VLANs, the backplane port must be tagged.)
- **Switch Fabric Module** —The central backplane aggregator for the system. To ensure cross-module communication (for example, within VLANs that span modules), you must configure the switch fabric module to include the VLANs that you configure on your switching modules. You configure the switch fabric module's ports in accordance with:
  - The chassis slot placement and VLAN configuration of your individual switching modules (configured through the EME).
  - The chassis slot placement of your GEN interface modules. To create VLANs for these non-local switching modules, you configure their VLANs by configuring the corresponding switch fabric module ports.

Features Your Switch 4007 supports the VLAN features listed in Table 48.

Table 48 VLAN Features

Feature	Layer 2 Modules and Switch Fabric Module	Multilayer Modules	Description
VLAN mode: allOpen or allClosed	Yes	Yes	On a per-module basis, establishes a less-restrictive VLAN environment (allOpen mode) or a more secure VLAN environment (allClosed mode). The VLAN mode dictates the requirements for the port-, protocol-, and network-based VLANs. See "VLAN allOpen or allClosed Mode" later in this chapter.
Per-port IEEE 802.1Q tagging	Yes	Yes	On a per-port basis, dictates that transmitted frames are encapsulated and tagged as specified in the IEEE 802.1Q standard and that received frames must be encapsulated and tagged. See the sections on port-, protocol-, and network-based VLANs later in this chapter for specific information on tagging for the different types of VLANs.
Port-based VLANS	Yes	Yes	Determine VLAN membership based solely on the port on which the frame was received. The system provides a special port-based VLAN by default, with all ports of all modules, called the *default VLAN.* The system also supports *static* VLAN configuration for both Layer 2 and Multilayer Switching Modules, and *dynamic* port-based VLAN configuration for Multilayer Switching Modules. See "User-Configured Port-based VLANs" and "Dynamic Port-based VLANs Using GVRP" later in this chapter for information on static and dynamic VLAN configuration.
Protocol-based VLANs	No	Yes	Determine VLAN membership based on the port on which the frame was received, as well as the protocol of the frame. You can use the protocol-based VLANs (and applied routing interfaces) to establish routing between VLANs. See "Protocol-based VLANs" later in this chapter. In addition to the user-defined protocol-based VLANs, the system supports a special type of protocol-based VLAN called a *router port IP VLAN*. This type of VLAN, which the system automatically generates when you define an IP interface as a router port IP interface, requires allClosed mode. See "VLANs Created by Router Port IP Interfaces" later in this chapter for more information.
Network-based VLANs (IP only)	No	Yes	Determine IP VLAN membership based on the port on which the frame was received, as well as the IP protocol and destination network address of the frame. See "Network-based IP VLANs" later in this chapter. .

Table 48 VLAN Features (continued)

Feature	Layer 2 Modules and Switch Fabric Module	Multilayer Modules	Description
Ignore STP mode	No	Yes, in allClosed mode	Ignores the blocking Spanning Tree Protocol (STP) mode for the ports of a designated VLAN. (One instance of STP runs on the module, but you can disable it on a per-VLAN basis.) This mode, only available in allClosed mode, is disabled by default. You select (on a per-VLAN basis), which VLANs ignore STP blocked ports. It is typically used for VLANs with router interfaces that ignore the STP state. This mode allows routing or bridging over a port that is blocked by STP. See "Ignore STP Mode" later in this chapter.

Key Concepts

Before you configure VLANs, review the following key concepts.

Related Standards and Protocols

The following standards and protocols apply to the VLANs that you can configure:

- IEEE 802.1Q is a standard for VLANs. It aims to:
  - Define an architecture to logically partition bridged LANs and provide services to defined user groups, independent of physical location
  - Allow interoperability between multivendor equipment.

  IEEE 802.1Q defines the bridging rules for VLANs (ingress and egress rules, as described in detail in "Rules of VLAN Operation" later in this chapter). The standard also specifies a tag format that embeds explicit VLAN membership information within each frame in a 12-bit VLAN ID (VID), that provides 4094 possible VLANs. IEEE 802.1D, which now incorporates 802.1p, uses this same frame format but it takes advantage of an additional 3 bits to specify the priority levels used for Class of Service differentiation.
- **Generic Attribute Registration Protocol (GARP)** — This protocol is defined in IEEE 802.1p, which is a supplement to the IEEE 802.1D standard. GARP is a Layer 2 transport mechanism that allows switches and end systems to propagate information across the switching domain.

- **GARP VLAN Registration Protocol (GVRP)** — This protocol, which is defined in IEEE 802.1Q, defines dynamic registration of VLANs that use IEEE 802.1Q tagging (the VLAN ID). GVRP is supported for this release on Multilayer Switching Modules.

Tagging Types

The system supports per-port tagging (that is, you can select IEEE 802.1Q tagging or no tagging on a per-port basis). Tagging and non-tagging ports can coexist in the same VLAN group.

- **Non-tagging mode** — The default tagging mode. Use this tagging mode for front-panel ports if the environment includes end stations that do not support 802.1Q VLANs. Non-tagged VLAN ports accept tagged frames; however, any traffic transmitted from an untagged port on a VLAN is untagged.
- **802.1Q tagging mode** — With this form of tagging, VLAN frames are encapsulated and tagged as specified in the IEEE 802.1Q standard. In frame tagging mode, an explicit header that identifies to which VLAN the frame belongs is inserted into each frame of interswitch data. Frames in the same VLAN can be tagged or untagged. An untagged port in a VLAN cannot insert a tag, but it can recognize a tagged frame. Use this mode for VLANs in an IEEE 802.1Q environment.

You must evaluate tagging for each switching module's front-panel ports and backplane ports as well as the switch fabric module ports:

- For front-panel ports, you must use tagging when you have ports shared by different VLANs (VLANs that overlap on ports) and there is no other distinguishing characteristic. For port-based VLANs, tagging must be used to distinguish the shared ports (only *one* VLAN's shared front-panel ports can be untagged; in all other VLANs, the shared ports must be tagged). For VLANs on Multilayer Switching Modules, tagging is required to differentiate between shared ports of the same protocol type and overlapped IP Layer 3 VLANs in allClosed mode.

- For backplane ports and switch fabric module ports, you must use tagging when these ports are shared by multiple VLANs. (Only *one* VLAN's backplane ports can be untagged; in all other VLANs defined across the backplane, the backplane ports must be tagged.) If you tag the backplane port of a switching module for a VLAN, you must also tag the corresponding switch fabric module port in that VLAN.

Devices (end stations, routers, switches, and so forth) that are connected to an explicitly tagged front-panel port must be capable of supporting 802.1Q tagging. If the front-panel port is untagged in the VLAN to which they belong, however, they do not have to support 802.1Q tagging.

VLAN IDs

Each VLAN is identified by its VLAN ID (VID). For VLANs that you create, the system keeps track of its used VLAN ID numbers to help you select the next available VLAN ID. Data frames sent by the system are tagged per IEEE 802.1Q (which contains the VID) if tagging is enabled on the transmit port for that VLAN. Tagged IEEE 802.1Q data frames that are received on the system are assigned to the VLAN that corresponds to both the VID contained in the tag and the protocol type.

Be aware of these additional guidelines:

- The default VLAN always uses the reserved VID of 1.
- Before assigning a VID, review the information in Table 49.

Table 49 Assigning ID Numbers to VLANs

VLAN ID Number	Description
VID 1	Reserved for the default VLAN assigned by IEEE and 3Com Corporation
VID 4095	Reserved
VID 2–4094	Numbers that you assign when you create VLANs

On Multilayer Switching Modules, if you rely on dynamic configuration to create a port-based VLAN based on GVRP updates, the VID is the unique IEEE 802.1Q VID.

If you define a router port IP interface, the system automatically creates a router port IP VLAN and assigns it the next available VID. See Chapter 16 for information on router port IP interfaces.

Terminology Review the following terms:

- **Default VLAN** — The predefined port-based VLAN interface on all switch fabric module ports and the ports of each switching module that always uses
  - VID 1
  - Protocol type unspecified (for Multilayer Switching Modules)
  - The name Default

  The default VLAN also initially includes all of the bridge ports without any tagging, but you can modify the bridge ports and tag status of the default VLAN. See "The Default VLAN" for more detailed information.
- **VLAN origin** — Whether the VLAN was created in one of the following ways:
  - **Statically** — The VLAN display shows an origin of `static` if you define the VLAN.
  - **Dynamically** — The VLAN display shows an origin of `GVRP` if the system learned the VLAN dynamically through GVRP.
  - **Router** — The VLAN display shows an origin of `router` if you have defined a router port IP interface on a single bridge port. When you define a router port IP interface, you must place the system in allClosed mode. This setting removes any allOpen VLANs and re-creates the default VLAN. See Chapter 16 for more information on defining router port IP interfaces.
- **Protocol suite** — On Multilayer Switching Modules, the protocol family that is associated with a protocol-based or network-based VLAN. The protocol suite is `unspecified` for the default VLAN and all port-based VLANs.
- **Layer 3 address** — On Multilayer Switching Modules, the network or subnetwork address that is associated with a network-based IP VLAN.
- **Port membership** — The bridge ports that you assign to be part of the VLAN. If you have created trunks, you must specify the anchor port (lowest-numbered) port in the trunk when you define the VLAN. All bridge ports are initially part of the default VLAN on each module. See Chapter 12.

- **VLAN name** — The name that you assign to the VLAN. It can contain up to 32 ASCII characters. If the name includes spaces, enclose the name in quotation marks. The default VLAN always uses the name Default.
- **Dynamic VLAN configuration** — The method that enables dynamic VLAN configuration of port-based VLANs and dynamic updates of IEEE 802.1Q tagged port-based VLANs, using the GARP VLAN Registration Protocol (GVRP).
- **Ingress and egress rules** — Ingress rules that determine the VLAN to which an incoming frame belongs. If it cannot be assigned to any VLAN, it is assigned to the null VLAN, which contains no ports and has no associated address table in allClosed mode. Egress rules determine whether the frame is forwarded, flooded, or filtered, as well as the tag status of the transmitted frame. For more information on ingress and egress rules, see “Rules of VLAN Operation” later in this chapter.

Key Guidelines for Implementation

Consider the following guidelines when you configure VLANs on your Switch 4007 system.

Migration Path for Network-based VLANs

On your multi-layer modules, you can either configure network-based IP VLANs or you can define a single VLAN with the protocol type IP and then define multiple IP routing interfaces fro that single protocol-based VLAN (an IP VLAN).

If you decide to convert an existing network-based VLAN to a protocol-based VLAN that has multiple interfaces associated with it, use the following procedure:

1. Remove any existing network-based VLANs on your Multilayer Switching Modules.
2. Define an IP VLAN or a VLAN that supports IP as one of its protocols.
3. Define multiple IP interfaces (with different IP addresses) to use that IP VLAN. (See Chapter 16.)

   You can define up to 32 IP interfaces on each Multilayer Switching Module, including IP routing interfaces for static VLANs, router port IP VLANs (described in the next section), or any combination of static VLANs and router port IP VLANs.

If you define multiple interfaces for an IP VLAN, you cannot subsequently modify that IP VLAN to supply Layer 3 address information. If only one routing interface is defined for the IP VLAN, then (at Release 3.0) you can supply Layer 3 address information as long as it matches the Layer 3 information specified for the routing interface.

This latter procedure is not recommended, because it makes the IP VLAN a network-based VLAN, which will not be supported at releases higher than 3.0.

If you continue to use network-based VLANs for Release 3.0 on your Multilayer Switching Modules, you are limited to defining only *one* IP routing interface for that VLAN. When you define an IP routing interface with the interface type vlan, the system does not allow you to select a network-based IP VLAN that already has a routing interface defined for it. For more information on IP routing interfaces, see Chapter 16.

VLANs Created by Router Port IP Interfaces

By default, the Multilayer Switching Modules use a routing over bridging model, in which any frame is bridged before it is potentially routed. If you want to define IP routing interfaces that use a routing versus bridging model, however, you can bypass your static VLAN configuration and instead go directly to defining an IP interface on a single router port (a router port IP interface).

If you define a router port IP interface, note the following information:

- Defining an IP interface for a router port requires the interface type port. Defining an IP interface for a configured IP VLAN requires you to specify the interface type of vlan.
- The IP interface definition procedure for a router port requires that you place the Multilayer Switching Module in allClosed mode. The allClosed mode prevents MAC addresses from being shared between the router port IP VLAN and any other VLANs and enables the router port to ignore Spanning Tree states on the port.
- After you define the router port IP interface and change the VLAN mode to allClosed, the following events occur:
  - The Multilayer Switching Module deletes all other VLANs and redefines the default VLAN. You must redefine any VLANs that you had configured, keeping in mind that unicast traffic will no longer be forwarded between VLANs. You must define routing interfaces to allow forwarding between VLANs. Also, you cannot specify the bridge port owned by the router port IP interface in any VLAN that you configure or modify.
  - The Multilayer Switching Module creates a special protocol-based VLAN called a router port IP VLAN and assigns to it the next available VID. The VLAN displays identify the origin of a router port IP VLAN as router, as well as the port that is owned by the router port IP interface. You cannot modify or remove a router port IP VLAN, nor can you change its Ignore STP mode (which is always enabled).
- To disable bridging entirely for the router port, remove that port from the default VLAN.

For more information about defining a router port IP interface on a Multilayer Switching Module, see Chapter 16.

Design Guidelines

- Before you create any VLANs, draw your chassis configuration and carefully identify how the VLANs that are associated with your modules are bridging (or, for Multilayer Switching Modules, bridging or routing). Remember that each VLAN constitutes a subnetwork. If a VLAN spans modules, each participating module must have its attached devices (and router interface, if applicable) configured for that subnetwork. Use your drawing to help you identify the configuration requirements for each port in a VLAN and to ensure that the switch fabric module is configured to pass traffic to the correct VLANs.
- To simplify your configuration, try to keep the number of VLANs as small as possible. Where possible, reduce the number of subnetworks.
- To ease configuration changes (such as moving a module to another slot while retaining the same VLAN configurations), you can use the staging option. To use this option, enable staging on the module by issuing the `module nvData staging` command and use the EME `staging` option to apply the configuration. You can also use EME commands to upload and download module configurations saved on a server. See the *Switch 4007 Enterprise Management Engine User Guide* for more information about the EME.
- If you lose track of your changes in a complicated VLAN configuration, it may be better to perform a nonvolatile data (nvData) reset operation than to make numerous VLAN changes. For example, you can use the `module nvData reset` command to return to a default module configuration, including the VLAN configuration. See Chapter 6 for more information.
- Evaluate whether you really need to use tagging on the front-panel ports of your switching modules. If you do need to tag front-panel ports in your user-configured VLANs, then your attached devices must be IEEE 802.1Q enabled.
- In general, tag the backplane ports of the switching modules and the corresponding switch fabric module ports when you define multiple VLANs that span modules (for bridging or routing). When multiple VLANs are defined across the backplane ports and switch fabric module ports, only *one* VLAN can be untagged and all others must be tagged.

(Example: If the backplane and switch fabric module ports for the default VLAN are untagged, the backplane and switch fabric module ports for all other VLANs must be tagged.) It is safer to tag the backplane and switch fabric module ports of *all* VLANs, although in some configurations, some overhead could be associated with tagging.

Procedural Guidelines

Follow these procedural guidelines to configure VLANs on the modules in your system:

1. Use the EME to connect to each Layer 2 and Multilayer Switching Module individually and configure the VLAN mode and VLANs for each module.
2. On each switching module, select the VLAN mode of allOpen or allClosed.
3. On each switching module, create the appropriate number of VLANs for your configuration. For each VLAN definition:
   - **a** Select a VID for the VLAN and provide information based on the type of VLAN: port-based information for Layer 2 modules; port-based, protocol-based, and network-based information for Multilayer Switching Modules.
   - **b** Include the appropriate front-panel ports. Tag the front-panel ports if you need to (that is, if the ports overlap with another VLAN and tagging is the only distinguishing characteristic). Remember that if you tag a port, the attached device must support IEEE 802.1Q tagging. If you are configuring a Multilayer Switching Module that serves as a router, your VLAN may or may not include front-panel ports.
   - **c** Include the backplane port of the switching module in the VLAN definition unless the VLAN traffic is limited to that switching module only and will not pass through the switch fabric module. If the switching module supports two backplane ports (and resides in a slot that supports two switch fabric module ports), you typically configure the lower-numbered backplane port. (Example: On a 10-port Layer 2 switching module, you configure port 11; on a 12-port Multilayer Switching Module, you configure port 13.) When you have multiple VLANs, tag the backplane port. (In a subsequent step, you must tag the associated switch fabric module backplane port as well.)

4 On each Multilayer switching module with VLANs that you want to perform routing, define a routing interface for each protocol-based or network-based VLAN. Verify that the routing interface is defined to use the same network or subnetwork as any other module that supports the VLAN.

5 Use the EME to connect to the switch fabric module and configure all VLANs that will pass traffic through the Layer 2 switch fabric module (that is, VLANs that are associated with switching modules or the GEN interface modules).

a For each VLAN definition that is associated with one or more switching modules, include the switch fabric module backplane ports that correspond to the VLAN's participating switching modules. (For example, if the VLAN's participating modules reside in slots 3 and 5, include switch fabric module ports 5 and 9 in the VLAN definition on the switch fabric module.) Tag these switch fabric module ports if the backplane ports of the corresponding switching modules are tagged. (For each VLAN, verify that the tagging type for a switch fabric module port matches its associated backplane switching module port.)

b For each VLAN definition for 2-port GEN interface modules, include the switch fabric module backplane ports that correspond to the VLAN's GEN interface modules. (For example, if the VLAN's GEN interface module resides in slot 6, you define switch fabric module ports 11 and 12 to be part of the VLAN.) Tag these ports if the front-panel ports of the GEN interface modules are connected to IEEEE 802.1Q enabled devices, such as other Switch 4007 systems or other 3Com switches.

Number of VLANs

You must evaluate the number of VLANs on a per-module basis. The module type determines the number of VLANs that can be supported:

- Each Layer 2 switching module supports a maximum of 127 port-based VLANs.
- Each Multilayer switching module use the following equation to determine the number of VLANs that are supported.

Equation for VLANs on Multilayer Switching Modules

To determine the number of VLANs of any type that you can have on a Multilayer Switching Module, use the following equation:

No._of_VLANs_supported = (125 / No._of_Protocol_Suites) minus 3

When you use the VLAN equation to calculate the number of VLANs that you may have on your Multilayer Switching Module, keep in mind that the formula provides only an estimate. You may see more or fewer VLANs, depending on your configuration, use of protocol suites, and chosen tag style. If, for example, you are using the Release 3.0 VLAN tag style of all ports, this formula generally yields a maximum; if you change to use the Release 1.2 tag style of taggedVlanPorts, then this formula generally yields a minimum number of VLANs.

A result of up to 64 is valid. If your result is greater than 64, you must observe 64 as the limit for the number of VLANs supported.

The number of allowable VLANs includes the default VLAN, and the number of protocol suites always includes the *unspecified* protocol type.

To perform the calculation, determine the total number of protocol suites used on your system. Remember to include the *unspecified* type for the default VLAN, even if you have removed the default VLAN and do not have other VLAN defined with the unspecified protocol type.

Use the following guidelines to count the protocol suites that are used on the Multilayer Switching Module:

- IP counts as one protocol suite for IP VLANs.
- AppleTalk counts as one protocol suite for AppleTalk VLANs.
- Generic IPX, which uses all four IPX types, counts as four protocol suites. (Each IPX type alone counts as one.) To conserve VLAN resources, it is better to specify a specific IPX frame type than to use generic IPX.

- DECnet counts as one protocol suite for DECnet VLANs.
- The unspecified type of protocol suite counts as one, whether or not the default VLAN or port-based VLANs are defined. Even if you have *only* the unspecified protocol suite on the system, the limit is still 64 VLANs.
- If you are using GVRP (for dynamic port-based VLANs), use the type unspecified in the VLAN formula.
- X25, SNA, Banyan VINES, and NetBIOS each count as one protocol suite for their respective VLANs.

In addition to the limit on the number of VLANs, a limit of 15 different protocols can be implemented by the protocol suites on the module. See Table 53 later in this chapter for a list of the supported protocol suites and the number of protocols within each suite.

The following examples show how to use the equation.

Example 1 You have 7 protocol suites on the Multilayer Switching Module (IP, AppleTalk, unspecified for the default VLAN, and generic IPX, which counts as 4 protocol suites):

(125 / 7) minus 3 = 14

In this configuration, the module supports a minimum of 14 VLANs. As shown in Table 53, these 7 protocol suites use 8 protocols (3 IP, 2 AppleTalk, 1 unspecified, and 2 generic IPX).

Example 2 You have 5 protocol suites: IP, unspecified, AppleTalk, IPX 802.2 Sub-Network Access Protocol (SNAP), and IPX 802.3 Raw:

(125 / 5) minus 3 = 22

In this configuration, the Multilayer Switching Module supports a minimum of 22 VLANs. As shown in Table 53, these 5 protocol suites use 7 protocols: 3 IP, 1 unspecified, 2 AppleTalk, 1 IPX 802.2 SNAP, and 0 IPX 802.3 Raw (because it does not use an Ethernet protocol type).

If you are upgrading from a Switch 4007 2.x release and the VLAN resource limit is reached during a power-on with a serial port console connection, you can use the Administration Console command `bridge vlan vlanAwareMode` *to change the VLAN aware mode to* `taggedVlanPorts`*. See the section "VLAN Aware Mode" next for more information.*

VLAN Aware Mode

For Multilayer Switching Modules only, VLAN aware mode accommodates the difference in VLAN resource usage as well as tagged-frame ingress rules between Release 2.x and Release 3.0. For more information on ingress rules, see "Rules of VLAN Operation" later in this chapter.

The VLAN aware mode, which you set with the Administration Console command `bridge vlan vlanAwareMode`, reflects the difference in VLAN resource usage and modes of tagging on Multilayer Switching Modules as follows:

- In Release 2.x, all bridge ports were *not* VLAN aware (tagging aware) unless they were assigned to a VLAN that has one or more tagged ports.
- In Release 3.0.0, all bridge ports become VLAN aware after a software update or after an NV data reset and do not have to be explicitly tagged to forward tagged frames.

This difference in resource usage and modes of tagging has the following impact: After you upgrade the system from 2.x to 3.0, the release uses VLAN resources differently than did Release 2.x and may cause a change in the total number of allowable VLANs.

VLAN aware mode is currently supported only through the Administration Console, not through Web Management or SNMP.

Initial installation of Release 3.0 provides a default VLAN aware mode of `allPorts`, which is consistent with the 3.0 ingress rules and resource allocation. If you upgrade your Multilayer Switching Module and the VLAN resource limit is reached during a power up with a serial port console connection, the console displays an error message similar to the following one to identify the index of the VLAN that it was unable to create:

```
Could not create VLAN xx - Internal resource threshold
exceeded
```

In this situation, the module removes all bridge ports from the VLAN that it could not restore from NV data, although it does maintain the previously stored NV data. To restore your VLANs after you see the resource error message, enter the `bridge vlan vlanAwareMode` command and then set the VLAN aware mode to `taggedVlanPorts`. If VLANs are already defined, the Administration Console prompts you to reboot the module to put the new mode into effect.

If you do not see the VLAN internal resource error message, maintain the default VLAN aware mode of `allPorts`. In this case, the module can accommodate the number of Release 2.x VLANs, but it now uses different ingress rules for tagged frames.

The Administration Console commands `bridge vlan summary` and `bridge vlan detail` display the current VLAN aware mode after the VLAN mode (allOpen or allClosed).

General Guidelines

- The VLAN mode of allOpen or allClosed applies to *all* VLANs that are associated with a switching module or the switch fabric module. (For Multilayer Switching Modules, this means VLANs with a static, dynamic, or router port origin). Configure the VLAN mode *before* you define any static VLANs. (As part of the configuration procedures for a router port IP interface on Multilayer Switching Modules, you must place the system in allClosed mode; see Chapter 16.

If you change the VLAN mode after you have defined VLANs, the interface module or switch fabric module deletes all configured VLANs and redefines the default VLAN. See "Modifying the VLAN Mode" later in this chapter.

- You can control STP settings as follows:
  - For all types of modules, you can enable or disable STP for the entire module, but not individual VLANs.
  - For all types of modules (regardless of the VLAN mode), you can enable or disable STP for individual ports by using a bridge port option (`bridge port stpState` on the Administration Console). See Chapter 9 for bridging information.
  - For Multilayer switching modules only, if you configure the module for allClosed mode, you can enable Ignore STP mode on a per-VLAN basis. See "Ignore STP Mode" for information on Ignore STP mode.
- On Multilayer Switching Modules, to take advantage of GVRP for dynamic configuration or dynamic updates of port-based VLANs, you must explicitly enable GVRP as both a bridge-wide parameters and a bridge-port parameter. See Chapter 9 for information about bridging parameters. See "Dynamic Port-based VLANs Using GVRP" later in this chapter for information about GVRP.

- You can configure overlapping VLANs as long as the VLANs have some distinguishing characteristic. For example, a bridge port can be shared by multiple VLANs as long as there is a distinguishing characteristic for the shared port (for example, for Multilayer Switching Modules: protocol type or tagging type; for Layer 2 modules: tagging type). In allClosed mode, you must tag the shared ports of any overlapped network-based VLANs.
- Consider maintaining the system's default VLAN. The default VLAN preserves the flooding of unspecified traffic because it initially contains all of the system's ports, with unspecified protocol information and no tagging.
- If you are using a Multilayer Switching Module to establish routing between static VLANs and configure a VLAN interface to support one or more routing protocols, configure the VLAN for the protocols *before* you configure a routing interface. For protocols other than IP, the module does not define the routing interface for a protocol if a VLAN for that protocol does not exist. If you define an IP interface and specify `vlan` as the interface type, the system does not define the IP routing interface unless you have an IP VLAN configured. See the appropriate routing protocol chapter for an overview of your routing options and guidelines. See Chapter 16 for information on defining either a IP router interface for a static IP VLAN or a router port IP interface.
- If you plan to use trunks (aggregated links), define the appropriate trunks *before* you define your VLANs. (If you define a VLAN with certain ports and subsequently configure some of those ports to be part of a trunk, the module removes those ports from the VLAN and places them in the default VLAN.) See "Trunking and the Default VLAN" later in this chapter for information about how trunking actions affect the default VLAN. When you define a VLAN that includes trunk ports, you must specify the trunk's anchor port (lowest-numbered port). For trunking information, see Chapter 12.
- When a frame is received, the frame is assigned to a VLAN using the ingress rules. See "Ingress Rules" later in this chapter. When it transmits the frame, the module determines the tag status (none or IEEE 802.1Q tagging) by referring to the tag status of the transmit port in the frame's assigned VLAN. In allOpen mode, a frame may be transmitted on a port that does not belong to the assigned VLAN, in which case, the frame is transmitted untagged.

VLAN allOpen or allClosed Mode

The VLAN mode affects the way in which a module address table is used. You can select allOpen or allClosed as the VLAN mode for the switch fabric module or for any Layer 2 or Multilayer switching module. The default is `allOpen`. VLAN modes on a module cannot be mixed; they must be either allOpen or allClosed for a module.

3Com's use of the term "allOpen" is equivalent to the IEEE Standard 802.1Q term "Shared VLAN Learning" (SVL). The term "allClosed" is equivalent to the IEEE 802.1Q term "Independent VLAN Learning" (IVL). 3Com imposes the restriction of choosing one VLAN mode per module; more complex logic for assigning SVL and IVL to individual ports is described in the IEEE 802.1Q standard.

Important Considerations

- In general, select your VLAN mode *before* you define your VLANs (VLANs with an origin of `static`).
- As part of the configuration procedures for a router port IP interface on a Multilayer Switching Module, you must place the system in allClosed mode. After you define a router port IP interface (and the system creates the router port VLAN), you cannot change the VLAN mode until you delete the router port IP interface. Select a VLAN mode as follows:
  - **allOpen** — Use this less restrictive mode if you have no security issues about the forwarding of data between VLANs. The allOpen mode is the default VLAN mode for all VLANs that you create. The allOpen mode implies that the module uses a single bridge address table for all of the VLANs on the module (the default configuration). If a device is a member of two VLANs, and the VLAN mode is `allOpen`, the address table discards the device address and learns the new address each time that the device transmits on the other VLAN.

    On Multilayer switching modules, allOpen mode permits untagged data with a unicast MAC address to be forwarded between VLANs, if the destination MAC address is in the forwarding table. Example: Untagged data received on IP VLAN 2 with a destination of IP VLAN 3 is forwarded there.

  - **allClosed** — Use this restrictive mode if you are concerned about security between VLANs. Data cannot be forwarded between VLANs (although data can still be routed between VLANs). The allClosed mode implies that each VLAN that you create has its own address table. If a device is a member of two VLANs, and the VLAN mode is `allClosed`, the device address is retained in both address tables. Unnecessary relearning by the address table is prevented, but data cannot be forwarded between VLANs. Router port IP interfaces require allClosed mode.

- If you are using allClosed mode and STP on a Multilayer Switching Module (with multiple routes to a destination), you can also specify the mode called *Ignore STP mode* to disable STP blocking for a specified static VLAN. (Although each VLAN has its own address table, there is only one STP on the module.) See "Ignore STP Mode" later in this chapter for information on this mode. (To disable STP blocking on a per-port basis with allOpen or allClosed VLANs, you can use the bridging option `bridge port stpState` on the Administration Console). See Chapter 9 for bridging information.

- Your selection of a VLAN mode affects how you manipulate bridge port addresses. Examples:

  - If you select allClosed mode, you *must* specify a VLAN interface index to identify the appropriate bridge address table.

  - If you select allOpen mode (the default), only one address table exists for the entire module, so you can manipulate the bridge port addresses without specifying a VLAN interface index.

VLANs on a module are either allOpen or allClosed. You select the VLAN mode on the Administration Console by using the `bridge vlan mode` command. Because each mode has its own set of rules, a Switch 4007 system may contain mixed modules of different modes. The chassis may contain VLANs on some modules that are *allOpen* and VLANs on other modules that are *allClosed*.

Modifying the VLAN Mode

To change your VLAN mode for a module:

1. For a Multilayer Switching Module, delete all routing interfaces (including router port IP interfaces). You cannot change the mode if you have router interfaces defined on the module.
2. Modify the VLAN mode to specify the new VLAN mode. When you change the mode, the module deletes all of your existing configured VLANs for the module and reverts to the default VLAN.
3. Reconfigure your VLANs (and, for Multilayer Switching Modules, redefine your routing interfaces).

Mode Requirements

Table 50 lists the requirements for defining VLANs in allOpen mode and allClosed mode.

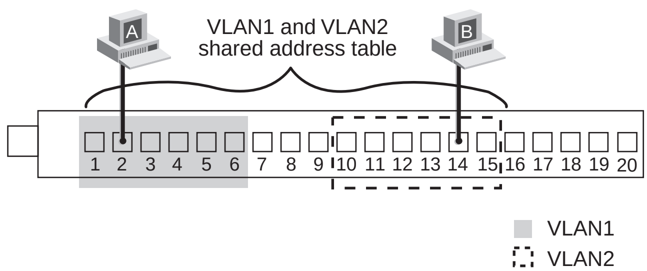
Table 50 Mode Requirements for Static VLANs

Type of Static VLAN	Requirements
Port-based (Layer 2 and Multilayer Switching Modules)	For *nonoverlapped* port-based VLANs: ■ Protocol type of unspecified ■ Separate member ports (each port-based VLAN owns a different set of ports) For *overlapped* port-based VLANs: ■ Protocol type of unspecified ■ IEEE 802.1Q tagging for shared ports (the shared ports can employ a tagging mode of none in only one VLAN; shared ports in all other VLANs must use IEEE 802.1Q tagging)
Protocol-based (Multilayer Switching Modules)	For *nonoverlapped* protocol-based VLANs: ■ Either the protocol type is unique per VLAN or the member ports are unique per VLAN For *overlapped* protocol-based VLANs (multiple VLANs of the same protocol type that share ports): ■ IEEE 802.1Q tagging for shared ports (the shared ports can employ a tagging mode of none in only one of the same protocol type VLANs; shared ports in all other VLANs of the same protocol type must use IEEE 802.1Q tagging
Network-based (IP VLAN only on Multilayer Switching Modules)	■ A Layer 3 address that is unique per network-based VLAN ■ For *allOpen mode*, no tagging restrictions on the shared ports ■ For *allClosed mode*, IEEE 802.1Q tagging for shared ports (the shared ports can employ a tagging mode of none in only one of the network-based VLANs; shared ports in all other network-based VLANs must use IEEE 802.1Q tagging

Using allOpen Mode

Figure 13 shows an allOpen configuration between two port-based VLANs that share the same address table. In this example, although Station-A and Station-B share a common address table, they must still adhere to broadcast containment rules. If Station-A sends a tagged frame with VLAN-2 in the tag, the frame is forwarded to Station-B if the VLAN is allOpen; if the VLAN mode is allClosed, the packet is not forwarded.

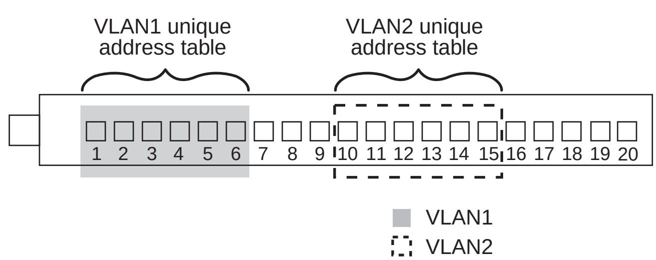
Figure 13 An allOpen Mode Configuration



Using allClosed Mode

Closed VLANs maintain their own unique address tables, as shown in Figure 14. For Layer 2 modules, port-based inter-VLAN traffic can be routed through a Layer 3 module. As shown in Figure 14, traffic can only be passed if the path is routed around VLAN-1 and VLAN-2.

Figure 14 An allClosed Mode Configuration



Port-based VLANs

Port-based VLANs logically group together one or more bridge ports on the module. On Multilayer Switching Modules, they use the generic protocol type *unspecified*. Each collection of bridge ports is designated as a *VLAN interface.* The VLAN interface belongs to a given VLAN. Flooding of all frames that are received on bridge ports in a VLAN interface is constrained to that VLAN interface.

Port-based VLANs group together one or more tagged or untagged bridge ports. The Switch 4007 supports the 802.1Q IEEE frame tagging standard on a per-port basis. The standard dictates that frames are encapsulated and tagged, which gives them a unique identification.

Each switching module (and the switch fabric module) supports the following types of port-based VLANs:

- The default VLAN, a special predefined VLAN
- User-configured port-based VLANs

In addition, Multilayer Switching Modules support dynamic port-based VLANs created using GVRP.

The Default VLAN

The system predefines a port-based VLAN to initially include all of the system's bridge ports without any tagging. For example, if you have four 10-port 100BASE-FX Fast Ethernet Layer 2 modules installed on your system, the default VLAN initially contains all 40 ports, plus the module backplane ports and the corresponding switch fabric module ports.

The default VLAN has the following properties:

- **VID** — 1
- **VLAN Name** — Default
- **VLAN Mode** — allOpen
- **Tag Type** — none (nontagging mode)

The default VLAN always uses a VID of 1, the name Default, and the protocol type unspecified (for Multilayer Switching Modules). No other VLAN can use a VID of 1.

This type of configuration has no restrictions on the flooding domain. You must set up your own VLANs to restrict the flooding domain.

Modifying the Default VLAN

The default VLAN is always associated with a VID of 1, the unspecified protocol type (for Multilayer Switching Modules), and the name `Default`. Initially, the default VLAN is also associated with all ports and no tagging. If necessary, you can modify the default VLAN on the modules in the system. For example, you may want to remove certain ports. Such a change does not prevent the system from adding a new module's bridge ports to the default VLAN.

The default VLAN is characterized by a VID of 1 and the unspecified protocol type. The following rules apply to the insertion of a new module:

- If you have modified the default VLAN to remove all ports, the ports of a newly inserted module are added to the default VLAN.
- If you have modified the default VLAN to tag a port, the ports of a newly inserted module are added to the default VLAN.
- If you have removed and subsequently redefine the default VLAN, the ports of a newly inserted module are added to the default VLAN.
- If you have removed the default VLAN and at least one other VLAN exists, the ports of a newly inserted module are not added to any VLAN.
- If you have removed the default VLAN and no other VLANs exist, a new default VLAN is created containing all ports when a new module is inserted.

To ensure that data can be forwarded, verify that a bridge port is associated with a VLAN. This association is mandatory in allClosed mode. If you remove the default VLAN (and you do not have other VLANs defined for the modules in the system), your ports may not forward data until you create a VLAN for them.

The default VLAN is the flood domain in any of the following situations:

- A module receives data for a protocol that is not supported by any VLAN on the module
- A module receives data for a protocol that is supported by defined VLANs, but these VLANs do not contain the port receiving the data.
- A module receives untagged data for a port and protocol that is supported by the switching module, but the port is tagged.

See "Rules of VLAN Operation" later in this chapter.

Trunking and the Default VLAN

Another benefit of maintaining the default VLAN (with any number of ports) involves trunking. 3Com strongly recommends that you define your trunks *before* you define your VLANs.

Trunking with the default VLAN intact

Trunking actions affect the default VLAN in the following ways:

- If you have only the default VLAN with all ports and you define a trunk (or subsequently remove a trunk), the ports listed in the VLAN summary for a module's default VLAN do not change. In this case, maintaining the default VLAN with all ports ensures that trunks can come and go without causing any VLAN changes.
- If you have the default VLAN as well as additional VLANs on a module and you subsequently define a trunk for ports in one of the other VLANs, the module removes those ports from that VLAN and places them in the default VLAN. The same action occurs when you remove an existing trunk from a VLAN that you created after the trunk. For example, on a 12-port Multilayer Switching Module:

Ports Before Action	Trunking Action	Ports After Action
default VLAN: ports 1-4 ipvlan1: ports 5–11	Define a trunk with ports 7, 8.	default VLAN: ports 1–4, 7, 8 ipvlan1: ports 5, 6, 9-11

- If you have the default VLAN as well as other VLANs on a module and you subsequently modify an existing trunk that has ports in one of the VLANs, any port that is removed from the trunk is removed from the VLAN and placed in the default VLAN. For example, on a 12-port Multilayer Switching Module:

Ports Before Action	Trunking Action	Ports After Action
default VLAN: ports 1–4 ipvlan1: ports 5–11 (Ports 5-8 are trunk ports.)	Modify existing trunk to have ports 6–8. (Remove port 5, the anchor port.)	default VLAN: ports 1–5 ipvlan1: ports 6–11 (Port 6 becomes new anchor port.)

Trunking with the default VLAN removed

If you remove the default VLAN, there is no place to return ports altered by trunking, as discussed in these examples:

- If you have VLANs (but no default VLAN) and you then define a trunk for ports in one of the VLANs, those ports are removed from that VLAN and are not assigned to any other VLAN. If you later remove the trunk, these ports are not reassigned to the VLAN; they no longer have a VLAN associated with them. For example, on a 12-port Multilayer Switching Module:

Ports Before Action	Trunking Action	Ports After Action
ipvlan1: ports 1–11	Define trunk with ports 5–8.	ipvlan1: ports 1–4, 9–11

- If you have VLANs (but no default VLAN) and you subsequently modify an existing trunk that has ports in one VLAN, any port removed from the trunk is removed from the VLAN and no longer has a VLAN. For example, on a 12-port Multilayer Switching Module:

Ports Before Action	Trunking Action	Ports After Action
ipvlan1: ports 1–11 (Ports 5-8 are trunk ports.)	Modify existing trunk to have ports 6–8. (Remove port 5, the anchor port.)	ipvlan1: ports 1–4, 6–11. (Port 6 becomes new anchor port.)

See Chapter 12 for more information on using trunks.

User-Configured Port-based VLANs

You can explicitly configure port-based VLAN interfaces on the Layer 2 and Multilayer switching modules as well as the switch fabric module.

Important Considerations

When you create this type of VLAN interface, review these guidelines:

- When you select the bridge ports that you want to be part of the VLAN, the bridge ports that you specify as part of the VLAN are the same as your physical ports, unless you have created trunks.
- If you define trunks, a single bridge port called the *anchor port* (the lowest-numbered port in the trunk) represents all ports that are part of the trunk. Only the anchor bridge port for the trunk is selectable when you are creating VLANs; the other bridge ports in the trunk are not selectable. For more information, see Chapter 12.
- Decide whether you want the ports that you are specifying for the VLAN interface to be shared by any other VLAN interface. Shared ports produce *overlapped* VLANs; ports that are not shared produce *nonoverlapped* VLANs.
- The per-port tagging options are IEEE 802.1Q tagging or no tagging. The IEEE 802.1Q tagging option embeds explicit VLAN membership information in each frame.
- Overlapped VLANs require tagging; that is, two port-based VLAN interfaces may contain the same bridge port if one of the VLAN interfaces defines the shared port to use IEEE 802.1Q tagging. This rule is true for either allOpen or allClosed mode. For example, a shared bridge port is set to tagging *none* for one VLAN and *IEEE 802.1Q* tagging for the other VLAN, or *IEEE 802.1Q* tagging for each VLAN.
- Multiple VLANs can span several modules, which may or may not overlap on the front-panel ports of the modules. However, they do overlap on the backplane ports. When multiple VLANs span modules, only one VLAN (usually, the default VLAN) can be untagged. Because all VLAN traffic flow between modules takes place through the GEN Switch Fabric Module by way of the backplane ports (regardless of the front-panel port configurations), the backplane ports for additional overlapping VLANs require explicit 802.1Q tagging.
- On Multilayer Switching Modules, port-based VLANs use the protocol type *unspecified*. This setting is implicit on a Layer 2 module. On a Multilayer Switching Module, you must specify the protocol type *unspecified* to create a port-based VLAN.

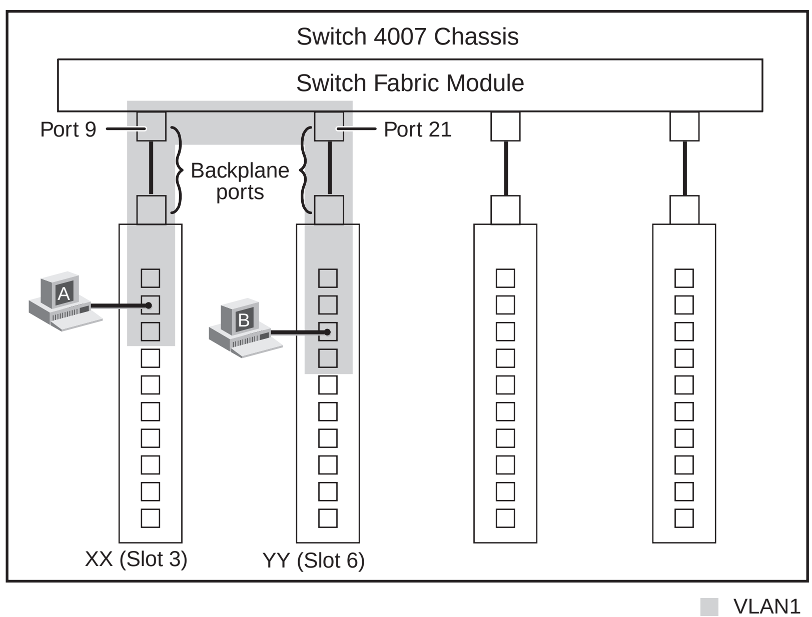
To define a port-based VLAN interface, specify this information:

- VID, or accept the next available VID.
- Bridge ports that are part of the VLAN. (If you have trunk ports, specify the anchor port for the trunk.)
- Protocol type *unspecified* (on Multilayer Switching Modules)
- Tag status (none or IEEE 802.1Q).
- Unique name of the VLAN interface.

Example 1: A Single VLAN Configuration

The configuration in Figure 15 shows a single VLAN (for example, a modified default VLAN) that spans two switching modules and pass traffic through the switch fabric module (which resides in slot 8 but is logically represented above the other modules).

Figure 15 Single VLAN Example



In this example:

- A single VLAN spans multiple switching modules. (It can be a modified default VLAN.)
- The backplane ports of the switching modules and the switch fabric module are part of the VLAN.
- All traffic that passes between switching modules flows through the switch fabric module:
  - The backplane port of Module-XX connects to Port 9 of the switch fabric module.
  - The backplane port of Module-YY connects to Port 21 of the switch fabric module.
- Station-A can pass traffic to Station-B.

Example 2: VLANs with Tagged Backplane Ports

The configuration in shows two VLANs that span two Layer 2 switching modules and pass traffic through the switch fabric module (which resides in slot 7 but is logically represented above the other modules):

- VLAN1 (the default, user-modified)
- VLAN2 (user-configured, port-based VLAN)

Because VLAN1 and VLAN2 span switching modules, they must be defined on the switch fabric module. One VLAN (VLAN1) must be tagged on the backplane ports of the switching modules and on the corresponding switch fabric module ports.

Figure 16 Two VLANs with Tagged Backplane Ports

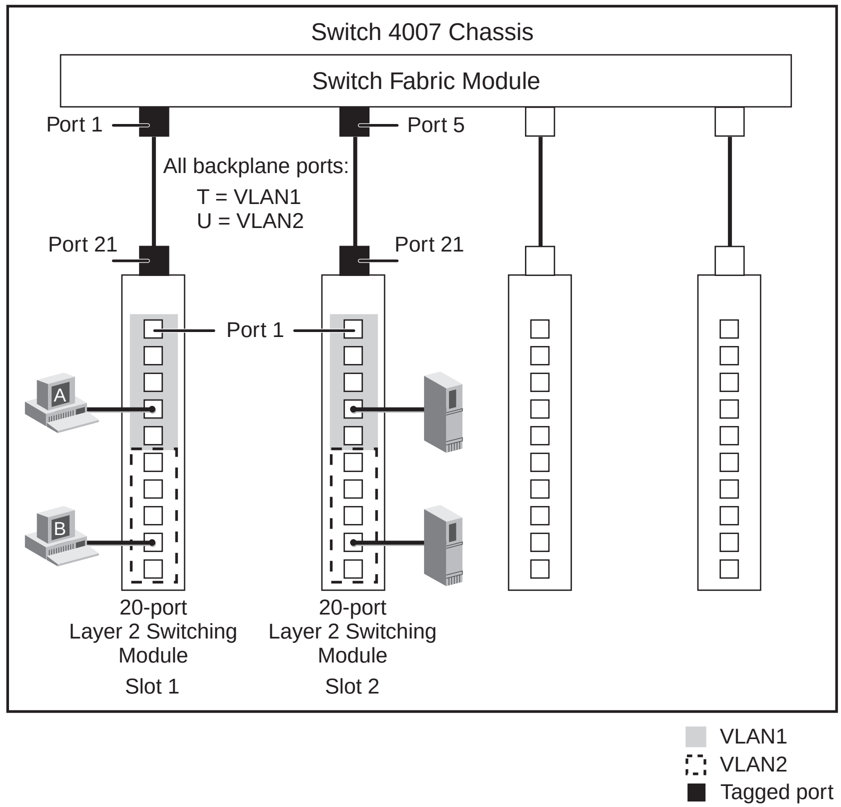


Table 51 lists the VLAN definitions for these port-based VLANs:

Table 51 Port-based VLANs with Tagged Backplane Ports

Slot 1 Module	Slot 2 Module	Switch Fabric Module
VLAN1 (default):	*VLAN1 (default):*	*VLAN1 (default):*
■ VLAN Index 1	■ VLAN Index 1	■ VLAN Index 1
■ VID 1	■ VID 1	■ VID 1
■ Ports 1–5, 21/22	■ Ports 1–5, 21/22	■ Ports 1,5
■ Tagging *none* front-panel ports 1–5	■ Tagging *none* front-panel ports 1–5	■ Tagging *802.1Q* fabric ports 1,5
■ Tagging *802.1Q* backplane port 21/22	■ Tagging *802.1Q* backplane port 21/22	

Table 51 Port-based VLANs with Tagged Backplane Ports (continued)

Slot 1 Module	Slot 2 Module	Switch Fabric Module
VLAN2:	*VLAN2:*	*VLAN2:*
■ VLAN Index 2	■ VLAN Index 2	■ VLAN Index 2
■ VID 20	■ VID 20	■ VID 20
■ Ports 6–10, 21/22	■ Ports 6–10, 21/22	■ Ports 1,5
■ Tagging *none* front-panel ports 6–10	■ Tagging *none* front-panel ports 6–10	■ Tagging *none* fabric ports 1,5
■ Tagging *none* backplane port 21/22	■ Tagging *none* backplane port 21/22	

Example 3: VLANs with Tagged Front-Panel Ports

The configuration in Figure 17 shows multiple overlapping VLANs that span two 20-port Layer 2 switching modules and pass traffic through the switch fabric module (which resides in slot 7 but is logically represented above the other modules).

In this example:

- There are two user-defined VLANs: VLAN2 and VLAN3. For the purposes of this example, assume that the default VLAN exists but is tagged on all ports that are shared by one or both of the user-configured VLANs. (You can also remove the shared ports from the default VLAN or remove the default VLAN.)
- VLAN2 and VLAN3 overlap on both the front-panel and backplane ports of Module-YY and on the switch fabric module backplane port (Port 17) that is connected to Module-YY.
- Because the membership of both VLANs is port-based, the shared ports (on both the front-panel and backplane ports) must be explicitly tagged.
- Station-E must support tagging because it is connected to a tagged port.
- The two overlapped front-panel ports on Module-YY can receive frames that are flooded on VLAN2 from Station-A, Station-B, and Station-E, or on VLAN3 from Station-C, Station-D, and Station-E.

  This communication is accomplished through the switch fabric module, which inserts an IEEE 802.1Q tag into the frame that contains the appropriate VLAN-ID. It then forwards the frame through its backplane port (Port 17) to Module-YY.

When the backplane port of Module-YY receives the frame, the tag identifies and knows to which VLAN the frame belongs.

Figure 17 Multiple VLAN Example with Tagged Front-Panel Ports

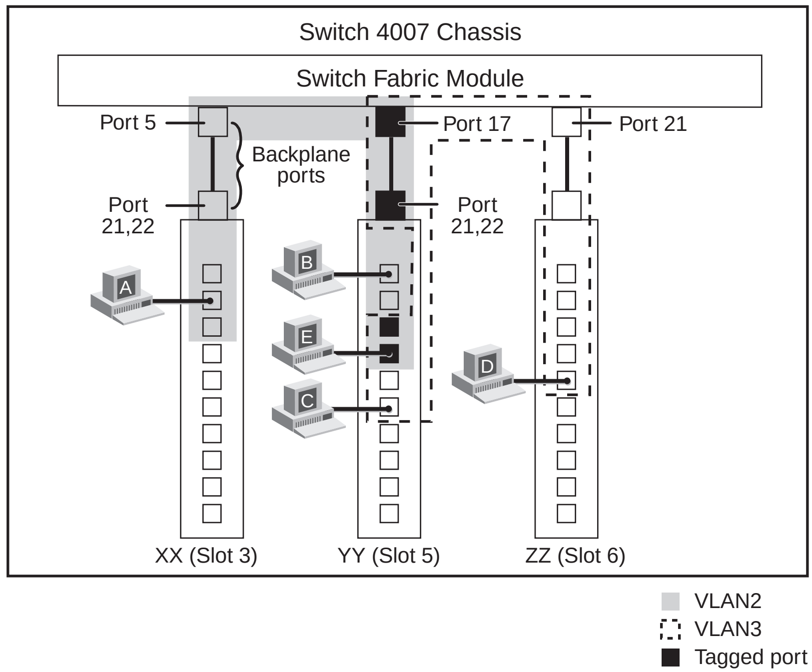


Table 52 lists the VLAN definitions for these port-based VLANs.

Table 52 Port-based VLANs with Tagged Front-Panel and Backplane Ports

Slot 3 Module	Slot 5 Module	Slot 6 Module	Switch Fabric Module
VLAN2: ■ VLAN Index 2 ■ VID 20 ■ Ports 1–3, 21/22 ■ Tagging *none* front-panel ports 1–3 ■ Tagging *none* for backplane port 21/22	*VLAN2:* ■ VLAN Index 2 ■ VID 20 ■ Ports 1–4, 21/22 ■ Tagging *none* front-panel ports 1,2 ■ Tagging *802.1Q* front-panel ports 3, 4 and backplane port 21/22	–	*VLAN2:* ■ VLAN Index 2 ■ VID 20 ■ Ports 5,17 ■ Tagging *none* port 5 ■ Tagging *802.1Q* port 17
–	*VLAN3:* ■ VLAN Index 3 ■ VID 30 ■ Ports 3–6, 21/22 ■ Tagging *802.1Q* front-panel ports 3, 4 and backplane port 21/22 ■ Tagging *none* front-panel ports 5,6	*VLAN3:* ■ VLAN Index 3 ■ VID 30 ■ Ports 1–5, 21/22 ■ Tagging *none* front-panel ports 1–5 ■ Tagging *none* backplane port 21/22	*VLAN3:* ■ VLAN Index 3 ■ VID 30 ■ Ports 17, 21 ■ Tagging *802.1Q* port 17 ■ Tagging *none* port 21

Dynamic Port-based VLANs Using GVRP

For Multilayer Switching Modules, GARP VLAN Registration Protocol (GVRP) can help you simplify the management of VLAN configurations in your larger networks.

GVRP allows the Multilayer Switching Module to:

- Dynamically create a port-based VLAN (unspecified protocol) with a specific VID and a specific port, based on updates from GVRP-enabled devices
- Learn, on a port-by-port basis, about GVRP updates to an existing port-based VLAN with that VID and IEEE 802.1Q tagging
- Send dynamic GVRP updates about its existing port-based VLANs.

GVRP allows your Multilayer Switching Module to advertise its manually configured IEEE 802.1Q VLANs to other devices supporting GVRP. Because the VLANs are advertised, GVRP-aware devices in the core of the network do not need manual configuration to pass IEEE 802.1Q frames to the proper destination. The method of VLAN advertisement used by all GVRP-capable switches involves protocol data units (PDUs), similar to the method used by STP. GVRP-capable devices send their updates to a well-known multicast address and all GVRP-capable devices listen to this address for information changes.

Enabling GVRP allows the Multilayer Switching Module dynamically adjust active network topologies in response to configuration changes in one or more VLANs. GVRP then advertises VLAN changes on each bridge to all other GVRP bridges in the network.

Important Considerations

To use GVRP, consider the following:

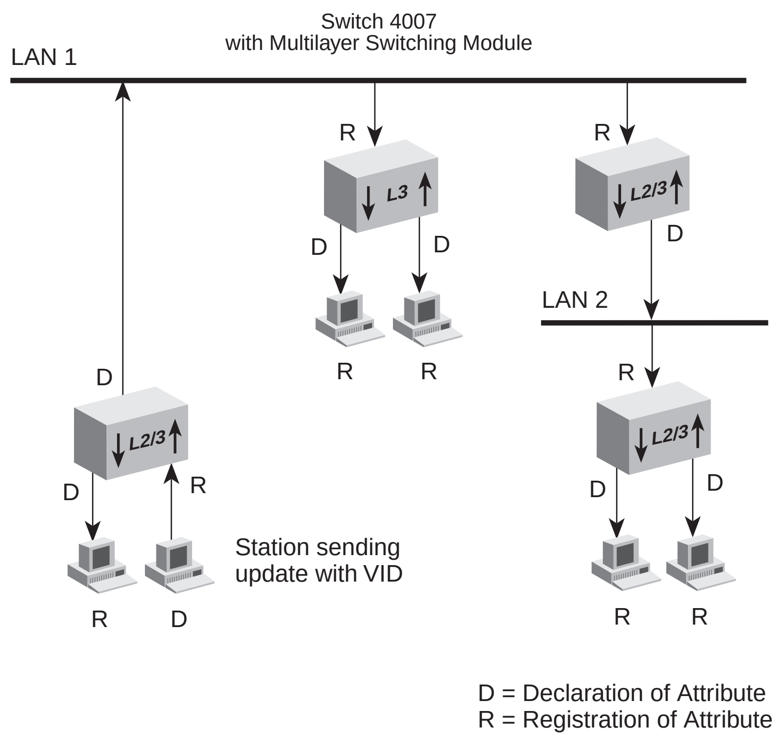
- You must explicitly enable GVRP as an entire bridge state and then as an individual bridge port state for the appropriate ports (see Chapter 9) to take advantage of dynamic IEEE 802.1Q VLAN configuration. By default, GVRP is disabled as both a bridge state and a bridge port state. If GVRP is enabled, the VLAN origin for a port-based VLAN is dynamic (with GVRP). When GVRP is disabled, the VLAN origin is either static (traditional static VLAN without GVRP) or router (router port).

- In a GVRP environment, devices must be GVRP-enabled (that is, support GVRP). These devices could be end stations with 3Com's Dynamic*Access*® software or other switches that explicitly enable GVRP.
- VLANs created dynamically with GVRP exist only as long as a GVRP-enabled device is sending updates. If the devices no longer send updates, or GVRP is disabled, or the module is rebooted, all dynamic VLANs are removed.
- GVRP updates are not sent out on any blocked STP ports. GVRP operates only on ports that are in the STP forwarding state. If GVRP is enabled, a port that changes to the STP forwarding state automatically begins to participate in GVRP. A port that changes to an STP state other than forwarding no longer participates in GVRP.
- The VLAN topologies that GVRP learns are treated differently from VLANs that are statically configured. GVRP's dynamic updates are not saved in NVRAM, while static updates are saved in NVRAM. When GVRP is disabled, the Multilayer Switching Module deletes *all* VLAN interfaces that were learned through GVRP and leaves unchanged all VLANs that were configured through the Administration Console, SNMP, or the Web Management software.
- GVRP manages the active topology, not nontopological data such as VLAN protocols. If you need to classify and analyze packets by VLAN protocols, you must manually configure protocol-based VLANs. But if the module needs to know only how to reach a given VLAN, then GVRP provides all necessary information.
- A GVRP-created VLAN is useful in situations where only Layer 2 switching needs to be performed for that VLAN. (Routing between a GVRP-created VLAN and another VLAN can be performed with an external router.) Because GVRP-created VLANs are assigned the unspecified protocol type, router interfaces cannot be assigned to them. Therefore, all communication within a GVRP-created VLAN is constrained to that VLAN in allClosed mode; in allOpen mode, only unicast frames with a known destination address can be transmitted to another VLAN.

Example: GVRP

Figure 18 shows how a GVRP update (with the VID) sent from one end station is propagated throughout the network.

Figure 18 Sample Configuration Using GVRP



Protocol-based VLANs

For Multilayer Switching Modules, protocol-based VLANs enable you to use protocol type and bridge ports as the distinguishing characteristics for your VLANs.

Important Considerations

When you create this type of VLAN interface, review these guidelines:

- If you plan to use the VLAN for *bridging* purposes, select one or more protocols per VLAN. Select them one protocol at a time.
- If you plan to use the VLAN for *routing*, you can select one or more protocols per VLAN, one protocol at a time, and subsequently define a routing interface for each routable protocol that is associated with the VLAN. You can perform routing as follows:
  - You can route between VLANs defined on Multilayer Switching Modules.
  - You can use a Multilayer Switching Module to route between VLANs that are defined on Layer 2 modules.
- The Multilayer Switching Modules support routing for three protocol suites: IP, IPX, and AppleTalk.
- To define a protocol-based VLAN interface, specify this information:
  - The VID, or accept the next-available VID.
  - The bridge ports that are part of the VLAN interface. (If you have trunk ports, specify the anchor port for the trunk.)
  - The protocol for the specified ports in the VLAN.
  - Tag status (none or IEEEE 802.1Q). IEEE 802.1Q tagging must be selected for ports that overlap on both port and protocol (for example, if two IPX VLANs overlap on port 3).
  - The name of this VLAN interface.
- If you use IP as the protocol and also specify a layer 3 address, the protocol-based VLAN becomes a *network-based VLAN*.

You can either configure network-based IP VLANs (IP VLANs with unique Layer 3 IP addresses) or you can define a single protocol-based VLAN with the protocol type IP and then define multiple IP routing interfaces for that VLAN. For more information on network-based VLANs, see "Network-based IP VLANs" later in this chapter.

Selecting a Protocol Suite

The protocol suite describes which protocol entities can comprise a protocol-based VLAN. For example, VLANs on the Multilayer Switching Module support the IP protocol suite, which has three protocol entities (IP, ARP, and RARP).

Table 53 lists the protocol suites that the Multilayer Switching Module supports, as well as the number of protocols that are associated with each protocol suite.

Table 53 Supported Protocol Suites for VLAN Configuration

Protocol Suite	Protocol Entities	Number of Protocol Suites	Number of Protocols in a Suite
IP	IP, ARP, RARP (Ethernet Version 2, SNAP PID)	1	3
Novell IPX	IPX (supports all of the following 4 IPX types):	4	2
	IPX - Type II (Ethernet Version 2)	1	1
	IPX - 802.2 LLC (DSAP/SSAP value 0xE0 hex)	1	0*
	IPX - 802.3 Raw (DSAP/SSAP value 0xFF hex)	1	0*
	IPX - 802.2-SNAP (DSAP/SSAP value 0xAA hex)	1	1
AppleTalk	DDP, AARP (Ethernet Version 2, SNAP PID)	1	2
Xerox XNS	XNS IDP, XNS Address Translation, XNS Compatibility (Ethernet Version 2, SNAP PID)	1	3
DECnet	DEC MOP, DEC Phase IV, DEC LAT, DEC LAVC (Ethernet Version 2, SNAP PID)	1	5
SNA	SNA Services over Ethernet (Ethernet Version 2 and DSAP/SSAP values 0x04 and 0x05 hexadecimal)	2	1
Banyan VINES	Banyan (Ethernet Version 2, DSAP/SSAP value 0xBC hexadecimal, SNAP PID)	1	1
X25	X.25 Layer 3 (Ethernet Version 2)	1	1
NetBIOS	NetBIOS (DSAP/SSAP value 0xF0 hexadecimal)	1	0*
Default (unspecified)	Default (all protocol types)	1	1

* This protocol does not use an Ethernet protocol type.

Your Multilayer Switching Modules impose two important limits regarding the number of VLANs and the number of protocols:

- **Number of VLANs supported** — To determine the minimum number of VLANs that the Multilayer Switching Module can support, use the equation described in "Number of VLANs" earlier in this chapter. A Multilayer Switching Module supports a maximum of 64 VLANs.
- **Maximum number of protocols** — Use the value 15 as the limit of protocols that can be implemented on the Multilayer Switching Module. A protocol suite that is used in more than one VLAN is counted only once towards the maximum number of protocols. For example, the DECnet protocol suite uses 5 of the available 15 protocols, regardless of the number of VLANs that use DECnet.

Establishing Routing Between VLANs

Your Multilayer Switching Modules support routing using IP, IPX, and AppleTalk VLANs. If VLANs are configured for other routable network layer protocols, they can communicate between them only via an external router or a Multilayer Switching Module configured for routing.

The Multilayer Switching Module's routing over bridging model lets you configure routing protocol interfaces based on a static VLAN defined for one or more protocols. You must first define a VLAN to support one or more protocols and then assign a routing interface for each protocol associated with the VLAN. (You can also opt to use a routing versus bridging model by defining a router port IP interface, as defined in Chapter 16).

Because the Multilayer Switching Modules support router port IP interfaces as well as IP router interfaces for static VLANs, you must now specify the interface type `vlan` *when you define an IP interface for a static VLAN.*

Important Considerations

To create an IP interface that can route through a static VLAN, you must:

1. Create a protocol-based IP VLAN for a group of bridge ports. (If the VLAN overlaps with another VLAN on any ports, be sure that you define it in accordance with the requirements of your VLAN mode.)

   (This IP VLAN does not need to contain Layer 3 information unless you want a network-based IP VLAN. See "Network-based IP VLANs" later in this chapter.)

2 Configure an IP routing interface with a network address and subnet mask and specify the interface type `vlan`.

3 Select the IP VLAN interface index that you want to *bind* to that IP interface.

If Layer 3 information is provided in the IP VLAN interface for which you are configuring an IP routing interface, the subnet portion of both addresses must be compatible. For example:

- IP VLAN subnet 157.103.54.0 with subnet mask of 255.255.255.0
- IP host interface address 157.103.54.254 with subnet mask of 255.255.255.0

Layer 2 (bridging) communication is still possible within an IP VLAN (or router interface) for the group of ports within that IP VLAN. For allClosed VLANs, IP data destined for a different IP subnetwork uses the IP routing interface to reach that different subnetwork even if the destination subnetwork is on a shared port. For allOpen VLANs, using the destination MAC address in the frame causes the frame to be bridged; otherwise, it is routed in the same manner as for allClosed VLANs.

4 Enable IP routing.

You perform similar steps to create IPX and AppleTalk routing interfaces. For more information, see the chapters in this guide for routing protocols such as IP, IPX, and AppleTalk.

Example 1: Routing Between Multilayer Modules

The configuration in Figure 19 shows routing between Multilayer Switching Modules. (The switch fabric module resides in slot 7 but is logically represented above the other modules.)

In this configuration:

- There are two Multilayer Switching Modules and the switch fabric module.
- There are four VLANs:
  - VLAN1, the default VLAN (tagged on backplane ports). The default VLAN is not represented in the figure.
  - VLAN2, an IP VLAN that is defined on the Multilayer Switching Module in slot 3.
  - VLAN3, an IP VLAN that is defined on the backplane ports of both Multilayer Switching Modules and a port-based VLAN defined on

the switch fabric module. The IP routing interfaces for IP VLAN 3 reside on the same subnet (33.3.3.0).

- VLAN4, an IP VLAN on the Multilayer Switching Module in slot 5.

- For this configuration to work, VLANs 2, 3, and 4 define IP routing interfaces, enable IP routing, and enable RIP.

Figure 19 Routing Between Two Multilayer Modules

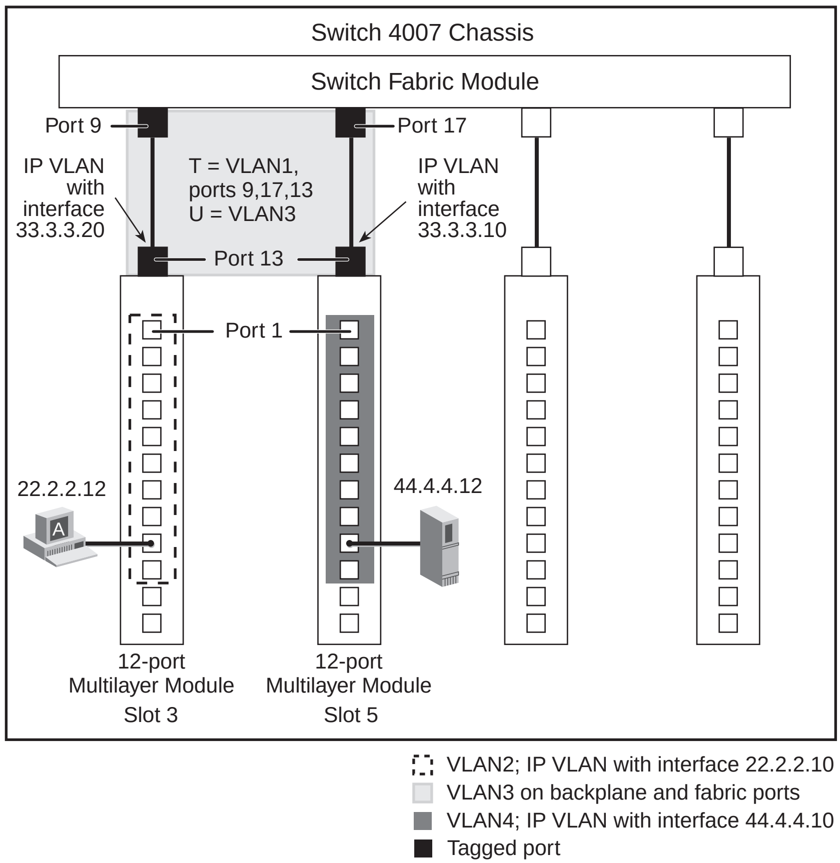


Table 54 lists the VLAN definitions for the modules in this configuration.

Table 54 Routing Between 2 Multilayer Modules over the Switch Fabric Module

Slot 3 Module	Slot 5 Module	Switch Fabric Module
VLAN1 (default): ■ VLAN Index 1 ■ VID 1 ■ Ports 1–12, 13 ■ Tagging *none* front-panel ports 1–12 ■ Tagging *802.1Q* backplane port 13	*VLAN1 (default):* ■ VLAN Index 1 ■ VID 1 ■ Ports 1–12, 13 ■ Tagging *none* front-panel ports 1–12 ■ Tagging *802.1Q* backplane port 13	*VLAN1 (default):* ■ VLAN Index 1 ■ VID 1 ■ Ports 9, 17 ■ Tagging *802.1Q* fabric ports 9, 17
VLAN2: ■ VLAN Index 2 ■ VID 20 ■ Ports 1–10 ■ Protocol type IP ■ No Layer 3 address ■ Tagging *none* front-panel ports IP router interface: 22.2.2.10	*VLAN4:* ■ VLAN Index 4 ■ VID 40 ■ Ports 1–10 ■ Protocol type IP ■ No Layer 3 address ■ Tagging *none* front-panel ports IP router interface: 44.4.4.10	–
VLAN3: ■ VLAN Index 3 ■ VID 30 ■ Port 13 ■ Protocol type IP ■ No Layer 3 address ■ Tagging *none* backplane port 13 IP router interface: 33.3.3.20	*VLAN3:* ■ VLAN Index 3 ■ VID 30 ■ Port 13 ■ Protocol type IP ■ No Layer 3 address ■ Tagging *none* backplane port 13 IP router interface: 33.3.3.10	*VLAN3:* ■ VLAN Index 3 ■ VID 30 ■ Ports 9, 17 ■ Tagging *none* fabric ports 9,17

Example 2: One-Armed Routing Configuration

Figure 20 shows a one-armed router configuration. (The switch fabric module resides in slot 7 but is logically represented above the other modules.)

In this configuration:

- There are three Layer 2 modules, a Multilayer Switching Module, and the switch fabric module.
- There are four VLANs:
  - VLAN1, the default VLAN on all modules (tagged on the front-panel and backplane ports of the Layer 2 modules and on the corresponding switch fabric module ports). The default VLAN is not represented in the figure.
  - VLAN2, a port-based VLAN that is defined on the Layer 2 modules in slot 1 and slot 2. It is defined as a protocol-based VLAN for IP on the Multilayer Switching Module in slot 4. The IP routing interface for IP VLAN 2 (150.10.2.12) is defined to be on the same subnetwork as the devices connected to the Layer 2 modules in slots 1 and 2. (For example, a PC is defined as 150.10.2.1.)
  - VLAN3, a port-based VLAN that is defined on the Layer 2 module in slot 2. It is defined as a protocol-based VLAN for IP on the Layer 3 module in slot 4. The IP routing interface for IP VLAN 3 (150.10.3.12) is defined to be on the same subnetwork as the devices that are connected to the Layer 2 module in slot 2.
  - VLAN4, a port-based VLAN that is defined on the Layer 2 module in slot 3. It is defined as a protocol-based VLAN for IP on the Layer 3 module in slot 4. The IP routing interface for IP VLAN 4 (150.10.4.12) is defined to be on the same subnetwork as the devices that are connected to the Layer 2 module in slot 3.
- On the Multilayer Switching Module, the backplane port 13 is part of all four VLANs and is tagged in all four. IP routing is enabled with RIP.

Figure 20 One-Armed Routing with Multilayer Module and Layer 2 Modules

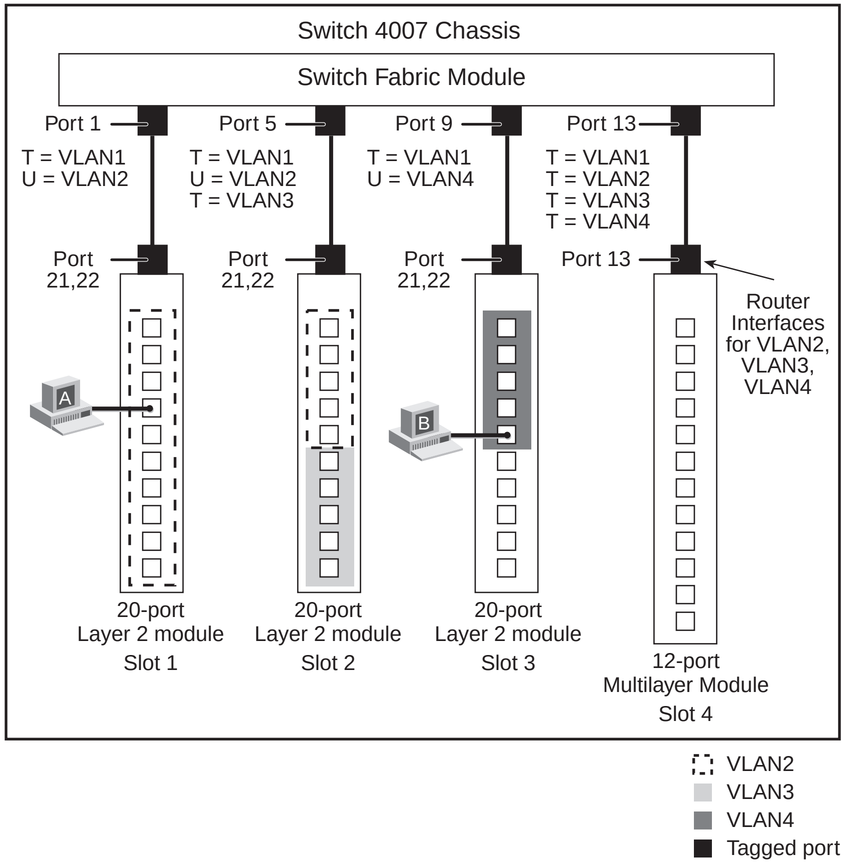


Table 55 defines the VLANs in this one-armed routing configuration.

Table 55 VLAN Definitions for One-Armed Routing Configuration

Slot 1 Layer 2 Module	Slot 2 Layer 2 Module	Slot 3 Layer 2 Module	Slot 4 Multilayer Module (Routing)	Switch Fabric (Layer 2) Module
VLAN1 (default): ▪ VLAN Index 1, VID 1 ▪ Ports 1 – 10, 21/22 ▪ Tagging *802.1Q* front ports 1–10, back ports 21/22	*VLAN1 (default):* ▪ VLAN Index 1, VID 1 ▪ Ports 1–10, 21/22 ▪ Tagging *802.1Q* front ports 1–10, back ports 21/22	*VLAN1 (default):* ▪ VLAN Index 1, VID 1 ▪ Ports 1–10, 21/22 ▪ Tagging *802.1Q* front ports 1–10, back ports 21/22	*VLAN1 (default):* ▪ VLAN Index 1,VID 1 ▪ Port 13 ▪ Tagging *802.1Q* backplane port 13	*VLAN1 (default):* ▪ VLAN Index 1, VID 1 ▪ Ports 1,5,9,15 ▪ Tagging *802.1Q* fabric ports 1,3,5,7
VLAN2: ▪ VLAN Index 2, VID 20 ▪ Ports 1–10, 21/22 ▪ Tagging *none* front-panel ports, backplane port 21/22	*VLAN2:* ▪ VLAN Index 2, VID 20 ▪ Ports 1–5, 21/22 ▪ Tagging *none* front-panel ports, backplane port 21/22	–	*VLAN2:* ▪ VLAN Index 2, VID 20 ▪ Port 13 ▪ Protocol type IP ▪ No Layer 3 address ▪ Tagging *802.1Q* backplane port 13 IP interface: 150.10.2.12	*VLAN2:* ▪ VLAN Index 2, VID 20 ▪ Ports 1,5,9 ▪ Tagging: *none* port 1,5 ▪ Tagging *802.1Q* port 13
–	*VLAN3:* ▪ VLAN Index 3, VID 30 ▪ Ports 6–10, 21/22 ▪ Tagging *none* front-panel ports ▪ Tagging *802.1Q* backplane port 21/22	–	*VLAN3:* ▪ VLAN Index 3, VID 30 ▪ Port 13 ▪ Protocol type IP ▪ No Layer 3 address ▪ Tagging *802.1Q* backplane port 13 IP interface: 150.10.3.12	*VLAN3:* ▪ VLAN Index 3, VID 30 ▪ Ports 5,13 ▪ Tagging *802.1Q* ports 3,7
–	–	*VLAN4:* ▪ VLAN Index 4, VID 40 ▪ Ports 1–5,21/22 ▪ Tagging *none* front ports, back port 21/22	*VLAN4:* ▪ VLAN Index 4, VID 40 ▪ Port 13 ▪ Protocol type IP ▪ No Layer 3 address ▪ Tagging *802.1Q* backplane port 13 IP interface: 150.10.4.12	*VLAN4:* ▪ VLAN Index 4, VID 40 ▪ Ports 9,13 ▪ Tagging *none* port 5; *802.1Q* port 13

Network-based IP VLANs

For IP VLANs only, you can also configure network-layer subnetwork addresses. With this additional Layer 3 information, you can create multiple independent IP VLANs with the same bridge ports. Untagged frames are assigned to a network-based VLAN according to both the protocol (IP) and the Layer 3 information in the IP header. Assigning Layer 3 address information to IP VLANs allows network administrators to manage their IP routing interfaces by subnetwork.

Network-based IP VLANs accommodate multiple routing interfaces over the same set of ports without tagging. Therefore, this option can be useful in allOpen mode. In allClosed mode, overlapped network-based IP VLANs must be IEEE 802.1Q tagged, which means that the system does not use the Layer 3 information.

Important Considerations

When you create this type of VLAN interface, review these guidelines:

- The network information is used only in situations where there are multiple network-based VLANs defined on a particular port. In situations where there is only one network-based VLAN defined on a port, the VLAN is treated as an ordinary IP protocol-based VLAN, and network-based information is ignored.
- When they are overlapped, network-based VLAN interfaces take precedence over protocol-based and port-based VLAN interfaces.
- You can define only *one* IP routing interface for a network-based VLAN. When you define an IP routing interface with the interface type `vlan`, the system will not allow you to select a network-based IP VLAN that already has a routing interface defined for it. For more information on IP routing interfaces, see Chapter 16.
- If you define multiple interfaces for an IP VLAN (instead of defining a network-based VLAN), you cannot subsequently modify that IP VLAN to supply Layer 3 address information. If only one routing interface is defined for the IP VLAN, then you can supply Layer 3 address information as long as it matches the Layer 3 information specified for the routing interface.
- In allClosed VLAN mode, you must also supply IEEE 802.1Q tagging for the ports (overlapped). Therefore, this feature has no added benefit. After IEEE 802.1Q tagging is implemented, implicit VLAN membership information such as the protocol or Layer 3 IP network address is not used, and the frame is assigned to the VLAN based solely on the tag VID and the receive port.

- In allOpen mode, you need not supply the IEEE 802.1Q tagging. However, to ensure line-speed throughput for overlapped network-based IP VLANs in allOpen mode, supply the IEEE 802.1Q tagging.

Example: Network-based VLANs

Figure 21 shows two IP network-based VLANs and two IPX protocol-based VLANs. (The switch fabric module resides in slot 8 but is logically represented above the other modules.)

In this configuration:

- There are two Multilayer Switching Modules and the switch fabric module. One module can handle servers; the other module can handle clients.
- There are four user-configured VLANs and the default VLAN:
  - VLAN1, the default VLAN on all modules, is untagged. The default VLAN is not represented in the figure.
  - VLAN2, a network-based VLAN for IP with Layer 3 address 22.2.2.0 on the Multilayer Switching Module in slot 3. The IP routing interface for IP VLAN 2 (22.2.2.10) on the Multilayer Switching Module in slot 3 is on the same subnet as the IP routing interface that is defined for VLAN2 on the Multilayer Switching Module in slot 5 (22.2.2.20).
  - VLAN3, a protocol-based VLAN for IPX-802.3 on the Multilayer Switching Module in slot 3. The IPX routing interface is defined to be on the same IPX network (1) as the IPX routing interface that is defined for VLAN3 on the Multilayer Switching Module in slot 5. Although there are overlapping ports in VLAN2 and VLAN3, the protocol type is the distinguishing factor.
  - VLAN4, a network-based VLAN for IP with Layer 3 address 44.4.4.0 on the Multilayer Switching Module in slot 5. The IP routing interface for IP VLAN 4 is 44.4.4.10.
  - VLAN5, a protocol-based VLAN for IPX-802.3 on the Multilayer Switching Module in slot 5. The IPX routing interface is defined to be on another IPX network (2).

Figure 21 Network-based VLANS

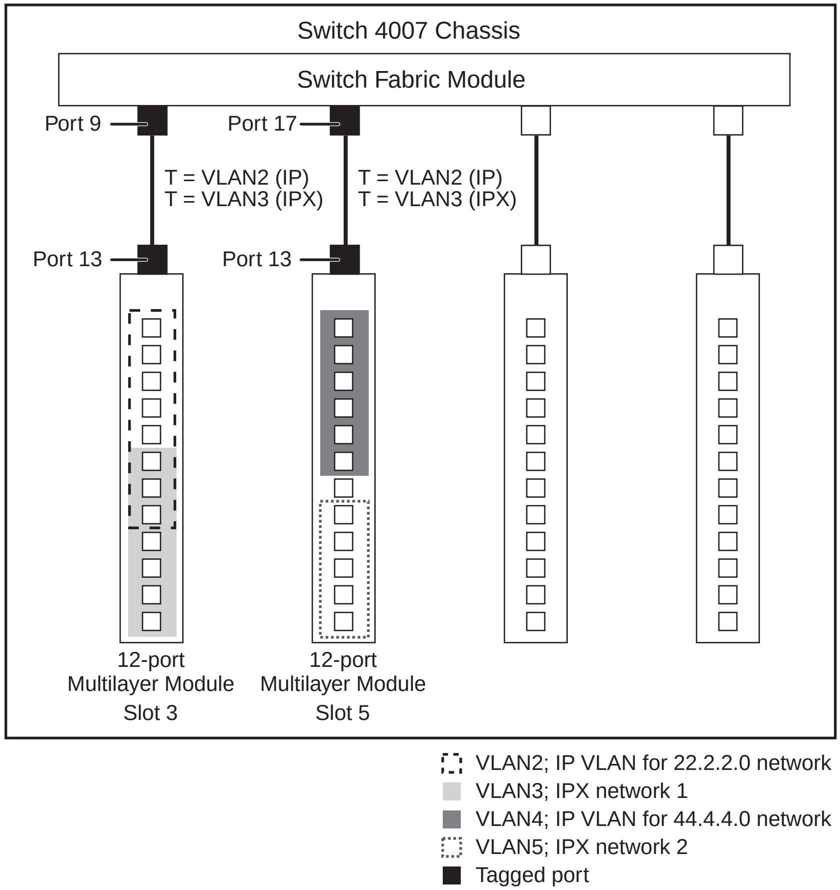


Table 56 defines the VLANs in this configuration:

Table 56 Network-based IP VLANs and IPX VLANs

Slot 3 Module	Slot 5 Module	Switch Fabric Module
VLAN2: ▪ VLAN Index 2, VID 20 ▪ Ports 1–8, 13 ▪ Protocol type IP ▪ 22.2.2.0 Layer 3 address ▪ Tagging *none* front-panel ports ▪ Tagging *802.1Q* backplane port 13 IP router interface: 22.2.2.10	*VLAN2:* ▪ VLAN Index 2, VID 20 ▪ Port 13 ▪ Protocol type IP ▪ 22.2.2.0 Layer 3 address ▪ Tagging *802.1Q* backplane port 13 IP router interface: 22.2.2.20	*VLAN2:* ▪ VLAN Index 2, VID 20 ▪ Ports 9,17 ▪ Tagging *802.1Q* fabric ports 9,17
VLAN3: ▪ VLAN Index 3, VID 30 ▪ Ports 6–12,13 ▪ Protocol type IPX- 802.3 ▪ Tagging *none* ports 6–12 ▪ Tagging *802.1Q* backplane port 13 IPX routing interface defined for IPX network 1	*VLAN3:* ▪ VLAN Index 3, VID 30 ▪ Port 13 ▪ Protocol type IPX-802.3 ▪ Tagging *802.1Q* backplane port 13 Router interface defined for same IPX network 1	*VLAN3:* ▪ VLAN Index 3, VID 30 ▪ Ports 9,17 ▪ Tagging *802.1Q* fabric ports 5,9
–	*VLAN4:* ▪ VLAN Index 4, VID 40 ▪ Ports 1–6 ▪ Protocol type IP ▪ 44.4.0 Layer 3 address ▪ Tagging *none* for front-panel ports IP router interface: 44.4.4.10	–
–	*VLAN5:* ▪ VLAN Index 5, VID 50 ▪ Ports 8–12 ▪ Protocol type IPX-802.3 ▪ Tagging *none* for front-panel ports IPX router interface defined for IPX network 2	–

Ignore STP Mode

When you use allClosed VLAN mode on a Multilayer Switching Module in your system, you can enable the module to ignore the Spanning Tree Protocol (STP) mode on a per-VLAN basis; that is, ignore STP blocked ports. (When STP detects multiple paths to a destination, it blocks all but one of the paths.)

If you have configured router port IP interfaces on your Multilayer Switching Modules (which causes the module to generate router port VLANs owned by the router IP interfaces), ignore STP mode is automatically enabled and you cannot disable it.

Important Considerations

- Ignore STP mode is disabled by default for static VLANs.
- You can use this mode *only* when the Multilayer Switching Module is in allClosed mode.
- Ignore STP mode is useful when you have redundant router connections between modules that have STP enabled. In this situation, if you want to create multiple VLANs and use one VLAN for routing, you can configure your module to ignore the STP blocking mode for that VLAN. This setting avoids disruptions to routing connectivity based on the STP state.
- To disable STP blocking on a *per-port* basis with allOpen or allClosed VLANs, use the bridging option (`bridge port stpState` on the Administration Console). See the Chapter 9 for bridging information.

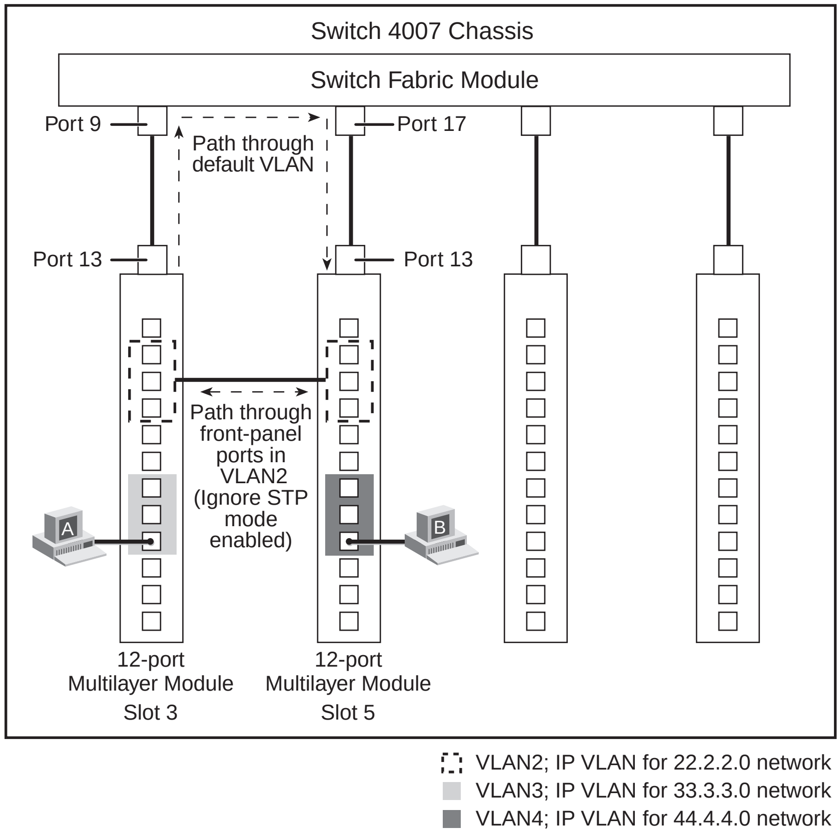
Ignore STP mode affects bridging as well as routing. If you have STP enabled on the system and you have redundant bridged paths between systems with different VLANs, STP blocks one of the paths unless you enable Ignore STP mode.

Example: Ignore STP Mode

Figure 22 shows two paths available through the default VLAN and IP VLAN2 if end station A wants to communicate with end station B. STP blocks the routed as well as bridged traffic for the one path through IP VLAN2 unless you enable Ignore STP Mode for IP VLAN2 (because the backplane ports have higher priority). With the blocking removed for IP routed traffic, the best path is used. (In this configuration, the switch fabric module resides in slot 7 but is logically represented above the other modules.)

IP VLAN2 has routing interfaces defined on both Multilayer Switching Modules (22.2.2.2 on the first Multilayer Switching Module and 22.2.2.3 on the second Multilayer Switching Module).

Figure 22 Ignore STP Mode



Rules of VLAN Operation

After you select a VLAN mode for your modules and create VLAN interfaces with VLAN characteristics such as IEEE 802.1Q or no tagging, port membership, protocol type, and Layer 3 (network) address information, the system determines the details of VLAN operation by observing two main types of rules:

- **Ingress rules** — Assign an incoming frame to a specific VLAN.
- **Egress rules** — Use standard bridging rules to determine whether the frame is forwarded, flooded, or filtered. These rules also determine the tag status of the transmitted frame.

These rules are classified in the IEEE 802.1Q standard. In addition, the system relies on some module-specific rules, discussed next.

Ingress Rules

These rules determine the VLAN to which an *incoming* frame belongs. The frame is assigned to the VLAN that has the most specific match. The system uses this protocol match hierarchy to find the most specific match.

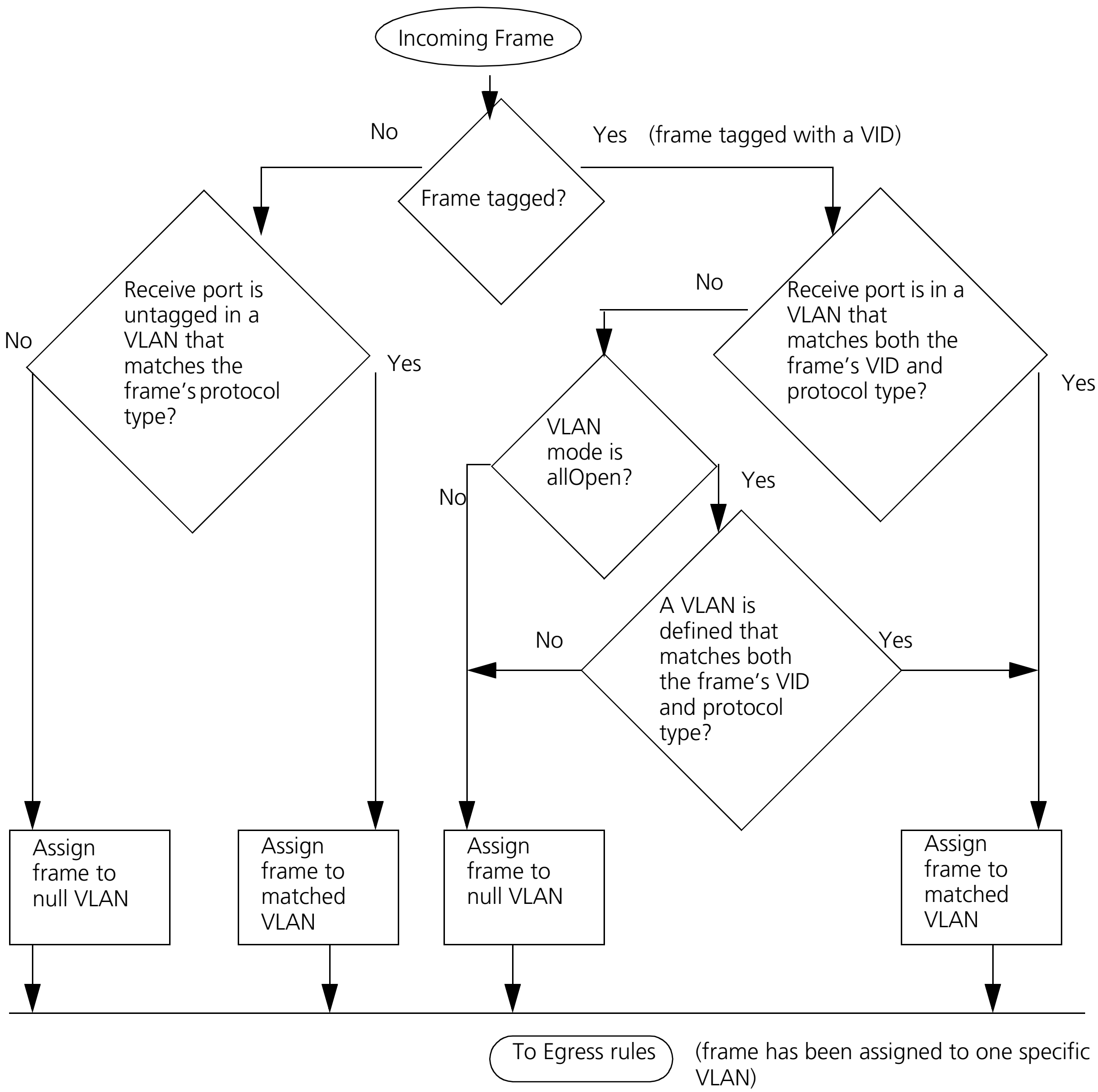
The ingress rules use the following hierarchy to determine the most specific match:

1. IEEE 802.1Q tag VID value.
2. For Multilayer Switching Modules, a specific protocol match (for example, IP, IPX, or AppleTalk).
3. The default VLAN (an untagged, unspecified protocol type VLAN with all ports and a VID of 1), or any VLAN that has the unspecified protocol type.
4. The *null VLAN*, a special VLAN that the system uses if the frame cannot be assigned to any VLAN. This VLAN has no ports and has no address table (in allClosed mode).

The Release 3.0 ingress rules are classified according to the tag status of the frame and the VLAN mode (allOpen for open VLANs or allClosed for closed VLANs). For the ingress rules, the system considers a priority tagged frame an untagged frame.

Figure 23 shows the flow chart for the Release 3.0 VLAN ingress rules for Multilayer Switching Modules.

Figure 23 Flow Chart for Release 3.0 Ingress Rules



The ingress rules for tagged frames also vary for the different releases. Table 57 summarizes the differences in ingress rules based on the releases.

Table 57 Ingress Rules for IEEE 802.1Q Tagged Frames Based on VLAN Mode and Release

VLAN Mode	Release 2.x	Release 3.0	Action Without Required Match
allOpen	The tagged frame is assigned to one of the configured VLANs if: ■ The VID of the frame matches that of a VLAN *and* ■ A port in that VLAN is tagged	The tagged frame is assigned to one of the configured VLANs if: ■ The VID of the frame matches that of a VLAN *and* ■ The protocol type of the frame matches that of the same VLAN	The frame is assigned to the null VLAN. It can still be forwarded (untagged) if the destination address of the frame is associated with another port in the bridge address table.
allClosed	The tagged frame is assigned to one of the configured VLANs if: ■ The receive port is in a VLAN with a VID that matches that of the frame *and* ■ A port in that VLAN is tagged.	The tagged frame is assigned to one of the configured VLANs if: ■ The receive port is in a VLAN with a VID that matches that of the frame *and* ■ The protocol type of the frame matches that of the same VLAN	The frame is assigned to the null VLAN and dropped.

Egress Rules

These rules determine whether the *outgoing* frame is forwarded, filtered (dropped), or flooded. They also determine the frame's tag status. The same standard bridging rules apply to both open and closed VLANs, but they result in different behavior depending on the allOpen mode (one address table for the module) versus allClosed mode (one address table for each VLAN). For example, on a Multilayer Switching Module, if a frame is associated with a VLAN that uses VID 1 and has a destination address associated with a VLAN that uses VID 2, the frame is flooded over the VID 1 VLAN in allClosed mode but forwarded untagged in allOpen mode.

Standard Bridging Rules for Outgoing Frames

The frame is handled according to these bridging rules:

- If the frame's destination address matches an address that was previously learned on the receive port, it is *filtered* (dropped).
- If the frame's destination address matches an address that was learned on a port other than the receive port, it is *forwarded* to that port.
- If a frame with an unknown, multicast, or broadcast destination address is received, then it is *flooded* (that is, forwarded to all ports on the VLAN that is associated with the frame, except the port on which it was received). See "Examples of Flooding and Forwarding Decisions" later in this chapter.
- If the frame's destination address matches a MAC address of one of the bridge's ports, or it matches an appropriate multicast address such as STP (if STP is enabled on the module), it is further processed and not forwarded immediately. This type of frame is either a management/configuration frame (such as a RIP update, SNMP get/set PDU, or an Administration Console Telnet packet), or it is a routed packet. If it is a routed packet, the Multilayer Switching Module performs the routing functions that is described in the appropriate routing chapter (for example, IP, IPX, or AppleTalk).

Tag Status Rules

After the VLAN and the transmit ports are determined for the frame, the Tag Status rules determine whether the frame is transmitted with an IEEE 802.1Q tag. For Multilayer Switching Modules, priority tagged frames for QoS use the same frame format as IEEE 802.1Q tagging but with a VID of 0. Priority tagged frames received by the Multilayer Switching Module are transmitted as either untagged frames (that is, no priority tagging) or IEEE 802.1Q tagged frames.

For each port on which the frame is to be transmitted, if that port is tagged for the VLAN associated with the frame, transmit the frame as a tagged frame; otherwise, transmit the frame as an untagged frame.

If the transmit port is not a member of the assigned VLAN, the frame is transmitted untagged. For VLANs in allOpen mode on Multilayer Switching Modules, this result may occur in either of these situations:

- *If the frame is assigned to the null VLAN. (The frame can still be forwarded if the address was statically entered in the address table or dynamically learned on another VLAN.)*
- *If the frame is assigned to a specific VLAN but the transmit port is* not *part of this VLAN.*

Examples of Flooding and Forwarding Decisions

This section provides several examples of flooding and forwarding decisions.

Example 1: Flooding Decisions for Protocol-based VLANs

Table 58 lists how flooding decisions are made according to three VLANs that are set up by protocol (assuming a 12-port configuration).

Table 58 Protocol-based VLANs and Flooding Decisions

Index	VLAN	Ports
1	Default	1–12
2	IP	1–8
3	IPX	9–11

Data received on this port	Is flooded on this VLAN	Because
IP - port 1	VLAN 2	IP data received matches IP VLAN on the source (receive) port.
IPX - port 11	VLAN 3	IPX data received matches IPX VLAN on the source port.
XNS - port 1	VLAN 1	XNS data received matches no protocol VLAN, so the Default VLAN is used.

Example 2: VLAN Exception Flooding

If data arrives on a bridge port for a certain protocol and VLANs for that protocol are defined in the module but not on that bridge port, the default VLAN defines the flooding domain for that data. This case is called *VLAN exception flooding.* Table 59 lists how the VLAN exception flooding decision is made (assuming a 12-port configuration).

Table 59 VLAN Exception Flooding

Index	VLAN	Ports
1	Default	1–12
2	IP	1–8

Data received on this port	Is flooded on this VLAN	Because
XNS - port 1	VLAN 1	XNS data on port 1 matches the unspecified protocol of the default VLAN on port 1.
IP - port 2	VLAN 2	IP data received matches IP VLAN 2 for source ports 1 - 8.
IP - port 12	VLAN 1	IP data on port 12 matches the unspecified protocol of the default VLAN on port 12.

Rules for Network-based (Layer 3) VLANs

Whenever an IP VLAN is defined with Layer 3 information, another VLAN is defined over the same ports called the *All IP Subnets* VLAN. Information about this VLAN is not available to the network administrator. Also, this VLAN has no VID associated with it and has no IEEE 802.1Q tagging on any of the ports. Incoming IP frames are assigned to this VLAN if they cannot be assigned to any of the network-based IP VLANs.

The following IP protocols are applicable to network-based VLANs:

- IP (hexadecimal 0800 or 0x0800)
- ARP (0x0806)
- RARP is (0x8035)

The frames that are associated with these protocols have different ingress rules for assignment to the appropriate network-based VLAN:

- **IP frames** — These frames are assigned to the network-based IP VLAN if the IP source address is consistent with the VLAN subnet and the IP destination address is one of the following:
  - 0.0.0.0
  - 255.255.255.255
  - A Class D (multicast) address

  Otherwise, assign to the network-based IP VLAN if the IP destination address is consistent with the VLAN subnetwork. Otherwise, assign to the *All IP Subnets* VLAN.
- **ARP frames** — These frames are assigned to the network-based IP VLAN if the IP destination address is consistent with the VLAN subnet and the IP source address is 0.0.0.0. Otherwise, assign to the network-based IP VLAN if the IP source address is consistent with the VLAN subnetwork. Otherwise, assign to the *All IP Subnets* VLAN.
- **RARP frames** — These frames are assigned to the *All IP Subnets* (Multicast) VLAN.

Example 3: Decisions for One Network-based VLAN

Table 60 lists the information for one network-based IP VLAN and how forwarding and flooding decisions are made for this VLAN.

Table 60 One Network-based VLAN and Forwarding/Flooding Decisions

Index	VID	VLAN Name	Ports	IP Subnet
2	2	IP_100	1 (untagged)	158.101.100.0
			2-6 (tagged)	mask: 255.255.255.0

Frame received on Port 1	Action
IP Frame (Protocol 0x0800), IP destination address (DA) 158.101.103.1, MAC DA is known on port 6	Frame is assigned to the IP_100 VLAN and transmitted on port 6 tagged.
RARP Response Frame (Protocol 0x8035), IP DA = 158.101.103.2, MAC DA is unknown	Frame is assigned to the IP_100 VLAN and transmitted on port 6 tagged.

Modifying and Removing VLANs

You can modify or remove any VLANs on the modules in your system. Review the following guidelines before you modify or remove VLANs:

- When you modify VLAN information for a VLAN interface on your module, you have the option to change VLAN characteristics such as the member bridge ports, protocol type, and form of explicit tagging.
- When you modify or remove a VLAN interface, you must specify a VLAN interface index to identify the VLAN interface. The Default VLAN always uses the VLAN interface index of 1. (The VLAN interface index is not the VID.)
- You cannot delete a VLAN for which you have defined a routing interface.
- If you add ports to a specific VLAN, you are permitting additional traffic through that port. If you remove ports from a specific VLAN and the Default VLAN is intact, those ports come under jurisdiction of the Default VLAN (unspecified protocol type, and no explicit or implicit tagging).
- Verify that each bridge port is associated with at least one VLAN in order to handle traffic.
- If you modify the Default VLAN to remove certain ports, verify that those ports are included in another VLAN. If the VLAN is in allClosed mode, those ports are not able to pass data if they are not part of another VLAN. See "Modifying the Default VLAN" earlier in this chapter for more information about the Default VLAN.
- If you remove the Default VLAN (and you have no other VLANs defined for the module), your ports may not be able to forward data until you create a VLAN for them (for example, if you are using allClosed mode).
- If you remove the Default VLAN, the system can no longer recognize any ports on a newly installed module, even if you delete the Default VLAN and then redefine it on the modules in the system.
- If you delete the Default VLAN, you must use the reserved VID of 1 if you redefine it.

Monitoring VLAN Statistics

When you display VLAN statistics on Multilayer Switching Modules, the module-generated statistics are valid only under either of these conditions:

- When the VLANs are defined for the same protocol type (or the type unspecified, for port-based VLANs) but do not have any overlapping ports (for example, an IP VLAN1 with ports 1–6 and IP VLAN2 with ports 7–12).
- If the VLANs are explicitly defined for different protocol types but may have overlapping ports (for example, an IP VLAN and an IPX VLAN that both use ports 2–4).

15 PACKET FILTERING

This chapter describes what packet filters are, how to create them, and how to use system utilities to apply them to ports of your Switch 4007 system. The chapter covers these topics:

- Packet Filtering Overview
- Key Concepts
- Important Considerations
- Managing Packet Filters
- Tools for Writing Filters
- Downloading Custom Packet Filters
- The Packet Filtering Language
- Common Syntax Errors
- Custom Packet Filter Examples
- Limits to Filter Size
- Using Port Groups in Custom Packet Filters
- Port Group Management and Control Functions
- Long Custom Filter Example

Packet filtering is supported on Multilayer Switching Modules only.

You can control and manage packet filters in either of these ways:

- *From the* `bridge packetFilter` *menu of the Administration Console. (See the* Switch 4007 Command Reference Guide.*) You can use the Administration Console after you log in to the system and connect to a slot that houses a Multilayer Switching Module.*
- *From the Filter Builder application in the Web Management software. The Filter Builder Help system serves as its documentation.*

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Packet Filtering Overview

The packet filtering feature allows a switch to make a permit-or-deny decision for each packet based on the packet contents. Use packet filters to control traffic on your network segments to:

- Improve LAN performance
- Implement LAN security controls
- Shape traffic flow to emulate virtual LAN (VLAN) behavior. See Chapter 14.

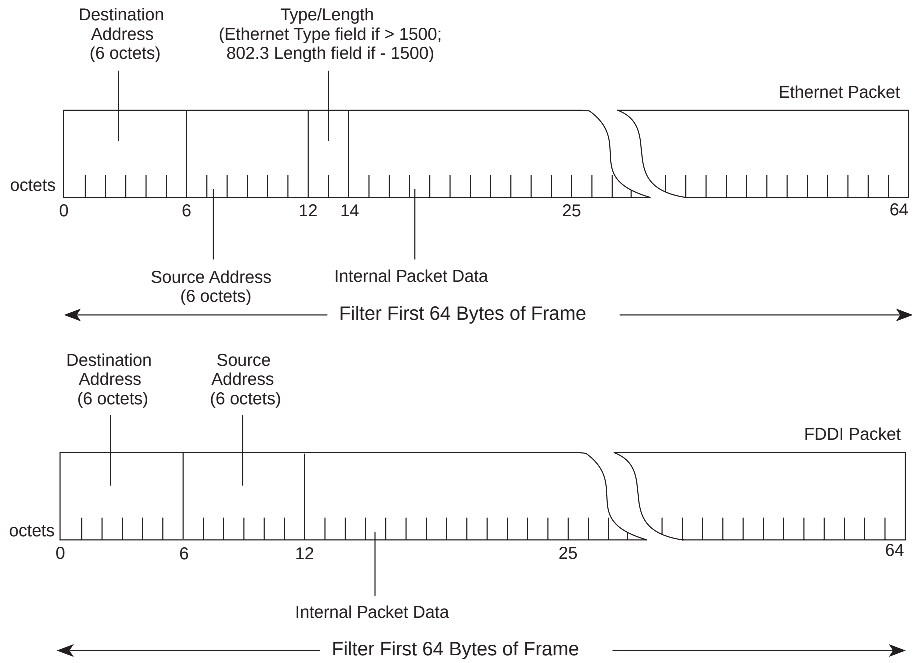
What Can You Filter?

Before you create a packet filter, you must decide which part of the packet you want to use for your filtering decisions. You can filter on any data in the first 64 bytes of the *frame*. You can filter Ethernet, Fast Ethernet, Fiber Distributed Data Interface (FDDI), or Gigabit Ethernet frames by the destination address, source address, type, length, or any attribute within the first 64 bytes. Keep in mind that the offsets may differ between FDDI and Ethernet packets, so the same filter may not work on all interfaces. Ethernet and FDDI packet fields are shown in Figure 24.

You can only filter Layer 2 traffic, not Layer 3 traffic. (This is true even though packet filtering is supported only on Multilayer Switching Modules.)

You must filter on the *input* packet type. For example, if you write a filter that you intend to assign to the transmit path of an Ethernet port, it will not be sufficient to compose a filter that only filters Ethernet traffic. This is because the filtering function is applied *before* the conversion to Ethernet format. Consider all possible sources of the packets. Might the packet originate as an FDDI packet? If so, then filter on the FDDI format as well as any Ethernet source formats.

Figure 24 Ethernet and FDDI Packet Fields



When Is a Filter Applied? — Paths

Packets travel on many different *paths* through the switch. You can control to which path a filter is applied.

Input Packet Filtering: Receive Path

Input packet filtering applies to packets immediately upon reaching the switch port, before they reach the switch's internal forwarding processing (*receive path*). Because the packets never enter the switch, the switch itself is protected against an external attack.

Output Packet Filtering: Transmit Path

Output packet filtering applies to packets after they have been through the switch's internal forward processing (*transmit path*).

Internal Packet Filtering: Receive Internal Path

Internal packet filtering applies to packets intended for the switch itself (such as pings, Telnet packets, and so forth) on the *receive internal path*.

Path Assignment

After you create a packet filter, you can assign it to any combination of the `transmit all`, `transmit multicast`, `receive all`, `receive multicast`, and `receive internal` paths of each port. The filter executes a series of operations on the packet's contents and, if the result is 0, it stops (filters) the packet. If the result is not 0, the filter allows the packet to pass.

The packet processing paths are defined in Table 61.

Table 61 Packet Processing Paths

Path	Description
Transmit all (txA)	All frames that are transmitted to the segment that is connected to the port
Transmit multicast (txM)	All multicast (including broadcast) frames that are transmitted to the segment connected to the port
Receive all (rxA)	All frames that are received by the port from the segment that is connected to the port
Receive multicast (rxM)	All multicast (including broadcast) frames that are received by the port from the segment that is connected to the port
Receive Internal (rxI)	All frames received by the port that have a system internal destination, such as ping and Telnet packets.

Key Concepts

Before you use packet filters, review the following key concepts and terms:

- **Standard Filters** — Packet filters that are supplied with the Switch 4007 that the hardware executes at wire speed. You can load them from the Administration Console, or select them from the set of predefined filters with the Filter Builder application. (Filter Builder is part of the Web Management suite of applications. See Table 63 later in this chapter.)

At present, one standard hardware filter is supported: the portGroup (rejdiffportgrp) filter.

- **Custom Filters** — Packet filters that are executed in software. You create custom filters in any of these ways:
  - By writing a filter definition using the filter definition language.
  - By selecting from among the predefined custom filters provided by the Filter Builder application
  - By using the Filter Builder's wizards to construct a new filter. (See Table 63 later in this chapter.)
- **Predefined filters** — Hardware and software filters that are supplied with the Filter Builder application. Filter Builder provides one standard filter that is executed by the hardware; the others are custom filters that are executed in software. (See Table 63 later in this chapter.)
- **Port Groups** — A collection of ports that you can reference in a packet filter. You create port groups from the Administration Console. You can specify different filtering rules between various port groups.

Standard Packet Filters

The Switch 4007 hardware supports standard packet filters. Standard filters are implemented in the ASIC hardware to achieve the wirespeed performance. To load them, use the Administration Console's `bridge packetfilter create` command.

At present, one standard hardware filter is supported: the portGroup (rejdiffportgrp) filter.

Standard packet filter support in the hardware is limited to the *receive all* and *transmit all* paths. Hardware filtering on the *receive multicast, transmit multicast,* and *receive internal* paths is not available; therefore, if you assign a standard filter to one of these paths, the module implements the filter in the software, which can affect performance.

Placing a filter on the *receive* path confines the packet to the segment that it originated from if it does not meet the forwarding criteria. Placing a filter on the *transmit* path prohibits a packet from accessing certain segments unless it meets the forwarding criteria. The module discards any packet that does not meet the forwarding criteria on the *transmit* path.

If you want to filter packets destined for the switch itself (for example, ping packets or Telnet packets), you must use the *receive internal* path. They are not filtered on the *receive all* path.

Custom Packet Filters

You create custom packet filters by writing a *packet filter definition*. Software implements custom filters. Consequently, use custom filters only on ports and paths that need them. Processing too many frames in software can affect performance on the ports where custom filters are assigned.

If you are trying to filter a certain type of broadcast or multicast packet assign the filter to either the TxM or the RxM paths, allowing only unicast traffic to bypass the filter.

Each packet-processing path on a port may have a unique custom packet filter definition or may share a definition with other ports on the module. Custom packet filter definitions are written in the *packet filter language,* which allows you to construct complex logical expressions.

After you write a packet filter definition, you load it onto a module; the corresponding port assignments are preserved in the nonvolatile memory (NVRAM) of the module, thus ensuring that the packet filter configuration for each module is saved across system or module reboots and power failures.

Important Considerations

- After you create a packet filter, you must:
  - Assign the filter to the applicable ports
  - Assign the filter to the applicable transmit and receive paths
  - Define port groups, if needed
- If you assign standard (hardware) filters on the receive multicast and transmit multicast paths, they will be executed in software which can slow the switch substantially. See "Standard Packet Filters" earlier in this chapter for details.
- Processing too many frames in software can affect performance on the ports where custom filters are assigned. See "Custom Packet Filters" earlier in this chapter for details.
- Exit a filter as soon as possible. See "Implementing Sequential Tests in a Packet Filter" later in this chapter for details.

Managing Packet Filters

You can control and manage packet filters from the `bridge packetFilter` menu of the Administration Console, as described in the *Switch 4007 Command Reference Guide*.

- **Listing packet filters** — You can list the packet filters that are defined for the module. The display includes the filter identification, filter name (if any), and filter assignments. Use the `bridge packetfilter list` command.
- **Displaying packet filters** — When you display the contents of a single packet filter, you select the packet filter using the filter id number that you see when you list the packet filters. The module displays the packet filter instructions. Comments in the original packet filter definition file are not displayed because they are not saved with the packet filter. Use the `bridge packetfilter display` command.
- **Creating packet filters** — You can create the standard (portGroup) hardware filter or your own custom packet filters. Placing a filter on the *receive* path confines the packet to the segment it originated from if it does not meet the forwarding criteria. Placing a filter on the *transmit* path prohibits a packet from accessing certain segments unless it meets the forwarding criteria. The module discards any packet that does not meet the forwarding criteria. Use the `bridge packetfilter create` command.

- **Deleting packet filters** — Deleting a packet filter removes the filter from the module. A filter cannot be deleted if it is assigned. You must unassign the filter from any ports before you can delete the filter. Use the `bridge packetfilter delete` command.
- **Editing, checking, and saving custom packet filters** — You can use the built-in line editor to edit custom packet filters. After you save the custom packet filter, the software examines it for syntax errors. The module software does not allow you to assign the packet filter to a port until the filter is error-free. Use the `bridge packetfilter edit` command.

  You can also edit a packet filter using an ASCII-based text editor such as EMACS, vi, or Notepad.
- **Loading packet filters** — After you create custom packet filters using an external text editor or Filter Builder, you must download the filters using the TFTP file transfer protocol onto the system from the network host on which you created them. When you have loaded it, the packet filter definition is converted into the internal format that is used by the packet filter code. Use the EME's `download` command to transfer the filter to the Switch 4007, then the `bridge packetfilter load` command to transfer it from the EME to the Multilayer Switching Module.
- **Assigning packet filters** — When you assign a packet filter to one or more ports, you must select the ports and a processing path. For descriptions of the available packet processing paths, see Table 61 at the beginning of this chapter. Each path of each port can have only one packet filter assigned to it; however, you can assign a single packet filter to multiple paths and ports. Use the `bridge packetfilter assign` command.
- **Unassigning packet filters from ports** — To unassign a packet filter from one or more ports, the packet filter must have been assigned to at least one port. Use the `bridge packetfilter unassign` command.
- **Defining port groups** — Before you assign packet filters that refer to port groups, create the port groups. See "Defining Port Groups" later in this chapter for more information.

See the *Switch 4007 Command Reference Guide* for more information about using these commands and management functions.

Tools for Writing Filters

The following tools can be used to create packet filters.

- ASCII Text Editor
- Built-in Line Editor
- Web Management Filter Builder Tool

ASCII Text Editor

You can create a new custom packet filter using an ASCII-based text editor (such as EMACS, vi, or Notepad). By using an ASCII-based text editor on a networked workstation, you can create multiple copies of the packet filter definition, which you can then store and copy onto one or more systems from the workstation. This method also allows you to archive copies of filter definitions and put them under source code control.

Built-in Line Editor

You can create a new custom packet filter using the line editor that is built into the Administration Console. The built-in text editor provides a minimal set of EMACS-style editing functions that you can use to edit a packet filter definition one line at a time. A single line is limited to no more than 79 characters. The number of lines is limited only by available memory.

Because the built-in editor is deliberately limited in scope, this method is most suited to making small temporary changes to a running filter.

The built-in editor assumes a terminal capability no higher than a glass tty (that is, it does not assume an addressable screen). You can place any ASCII printable character into the editing buffer at the cursor position. If the number of characters in the line buffer exceed the maximum number of characters permitted for the line, the characters that fall outside maximum line length are discarded. The built-in editor initially operates in *insert* mode.

Table 62 summarizes the commands that the editor supports.

Table 62 Commands for the Built-In Packet Filter Editor

Command	Keys	Description
List buffer	Ctrl+l	Displays each of the lines in the editing buffer, and then redisplays the line currently being edited.
Next Line	Ctrl+n	Moves cursor to start of next line.
Previous Line	Ctrl+p	Moves cursor to start of previous line.
Start of Line	Ctrl+a	Moves cursor to the start of the line it is in.
End of Line	Ctrl+e	Moves cursor to the end of the line it is in.
Left 1 Character	Ctrl+b	Moves cursor *left* one character within a line.
Right 1 Character	Ctrl+f	Moves cursor *right* one character within a line.
Insert Line	Enter	Inserts a new line. The new line becomes the current line, with the cursor positioned at the start. If the cursor is positioned over the first character on a line when you press [Enter], a blank new line is inserted before the current line. Otherwise, the current line is split at the cursor position, with the current line retaining the characters before the cursor, followed by the new line containing the rest of the characters.
Delete Previous Character	Ctrl+h	Deletes a single character preceding the cursor and shifts the remainder of the line *left* one position.
Delete Current Character	Ctrl+d	Deletes a single character under the cursor and shifts the remainder of the line *left* one position.
Delete Line	Ctrl+k	Deletes the remainder of the line from the current cursor position. If the cursor is positioned over the first character, all of the characters on the line are deleted, but the line is retained. A second Delete Line command removes the line from the edit buffer.
Insert/Overstrike Toggle	Ctrl+o	Toggles between the insert mode and overstrike mode.
Write Changes	Ctrl+w	Writes (saves) the current contents of the edit buffer into the packet filter definition. No syntax verification of the definition is performed at this point other than to verify that the length of the source is within the maximum limits. If the source is too long, the message `Error: Edit buffer exceeds maximum length` is displayed. The contents of the edit buffer are unaffected; however, the packet filter definition contains only those lines that fit entirely within the length limitation.
Exit Editor	ESC	Allows you to leave the editor. You receive a warning if the edit buffer has not been successfully written since the last modification. You can either discard the changes or return to the editor. Note that only those changes made since the last Write Changes command are discarded.

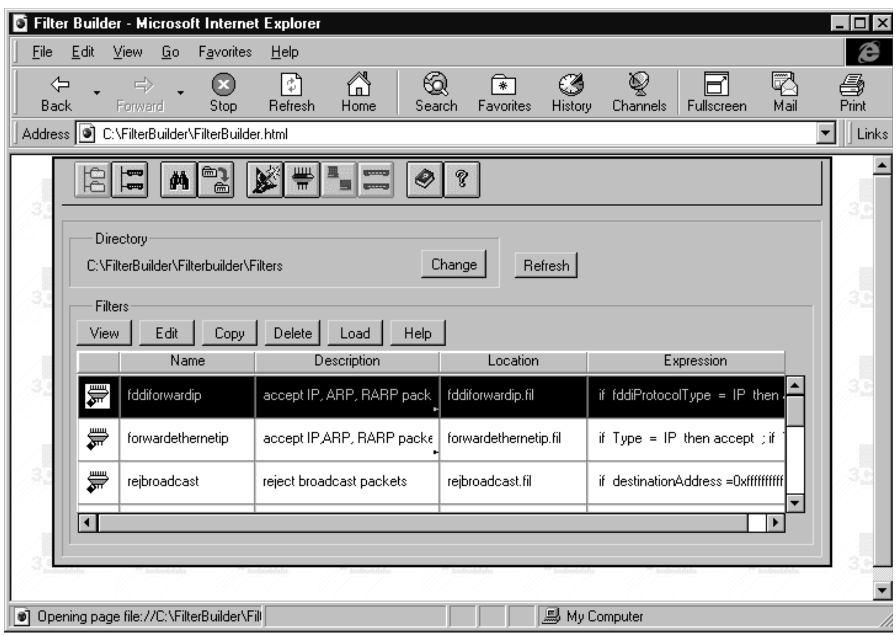
Web Management Filter Builder Tool

Filter Builder is part of the Web Management tool suite. You can use Filter Builder as a standalone application on your Unix or PC system to create your own custom filters and save them as ASCII files. You then download the files containing the custom filters to the switch using TFTP.

All Filter Builder functions that require an IP connection to the Switch 4007 are not supported in Release 3.0. This includes automatic filter downloads, defining port groups, and assigning filters to ports, port groups, and paths on the switch. Instead, connect to the switch manually to perform these functions through the Administration Console as described in the "Packet Filters" chapter of the Command Reference Guide*.*

With Filter Builder, you can implement custom packet filters easily and verify that your filters are syntactically correct before you test them on the module. Figure 25 shows the Filter Builder configuration form.

Figure 25 Filter Builder Configuration Form



Filter Builder includes 10 predefined filters, which are displayed on the Filter screen. Table 63 lists the filters by name, what each does, and whether the filter operates in the software or the hardware.

Table 63 Predefined Filter Builder Packet Filters

Filter Name	Type	Filtering Function	Implemented
fddiforwardip	Custom	Forwards FDDI IP, ARP, and RARP packets	Software
forwardethernetip	Custom	Forwards Ethernet IP, ARP, and RARP packets	Software
rejbroadcast	Custom	Rejects broadcast packets	Software
rejdiffaddgrp	Standard	Rejects packets from a specific address group	Software
rejdiffportgrp	Standard	Rejects packets from a specific port group	Hardware
rejethernetappletalk	Custom	Rejects Ethernet AppleTalk packets	Software
rejethernetipx802	Custom	Rejects Ethernet IPX packets	Software
rejfddiat	Custom	Rejects FDDI AppleTalk packets	Software
rejfddiipx802	Custom	Rejects FDDI IPX packets	Software
rejmulticast	Custom	Rejects multicast packets	Software

You can distinguish predefined filters from the custom filters that you create by the icon that pertains to each filter's name in the list on the Filter tab. The icon for predefined filters has a lock in the lower left corner, which indicates that the filter is write protected; you cannot edit or delete it.

Although the predefined filters are write-protected, you can edit a predefined filter indirectly *by copying it, giving it a new name, and then editing it.*

To create a filter, Filter Builder has two interfaces:

- **Filter Wizard** — If you are unfamiliar with the packet filtering or to create a simple filter, use this interface.
- **Create or Edit Filter window** — If you are familiar with the packet filtering or to create a complex filter, use this interface.

  For more information on the Filter Builder tool, see the *Web Management User Guide* and the Filter Builder's Help system.

Downloading Custom Packet Filters

Downloading a packet filter to the Switch 4007 is a two-step process. You log in to the EME and enter the `download module` command with a file type of `filter`. The EME does a TFTP file transfer and sends the file to the module. The module places the file in a temporary buffer. You then connect to the module and enter the `bridge packetFilter load` command to store the filter on the module. This command is not interactive because the module knows where to find the filter.

Because this is a two-step process there are restrictions placed on each step of the process. After you download a filter to a module, you cannot download a second filter until the first filter is stored. A module only has one buffer to hold one unstored filter at a time. If you attempt a second download, an error message is displayed and the transfer is aborted. If you download a filter file and then do not perform the second part of the process to store the filter, the filter file will be lost upon reset of the module. The filter is stored in a buffer on the module and not in NVRAM until it is loaded. It is not saved when a module is reset.

Setting Up Your Environment

Before you attempt to load a Filter Builder predefined or custom filter on a device, you must have the following running on your PC:

- The latest version of Filter Builder.

  Filter Builder is part of the Web Management suite of applications on the Switch 4007 software CD. See the *Web Management User Guide* for procedures and software prerequisites.

- A TFTP server application.

  This application must be set up with access to the appropriate IP address for:

  - The PC where Filter Builder is running, and
  - The Switch 4007 on which you are trying to load the filter.

  TFTP's root directory must be configured with the path where filters are stored, or the filter must be copied to TFTP's configured root directory.

Loading a Custom Filter on the Switch 4007

Here are step-by-step directions to load the filter on an EME, and then to load and assign filters on the Multilayer Switching Module:

1 Log in to a Switch 4007.

2 At the EME's prompt, enter a `download` command using the following syntax:

```
download module <slot.subslot> filter <IP address>
   <filterfilename>
```

For example:

```
CB9000>download module 6.1 filter 159.101.69.20 rejfddiat.fil
File transfer request pending.
Downloading file from external file server to eme - 000000289
Downloading file from eme to module 6.1 - 000000289
File transfer completed successfully.
```

The predefined filters that come with Filter Builder are found in the /3Com/Filterbuilder/Filters directory, which is the default directory for Filter Builder when installed from WebManage.exe.

3 Connect to the module:

For example:

```
CB9000> connect 6.1
```

4 At the module prompt, load the filter:

For example:

```
CB9000@slot6.1 [12-E/FEN-TX-L3] (): bridge packetFilter load
Packet filter 1 stored.
```

5 Verify that the filter has been loaded:

For example:

```
CB9000@slot6.1 [12-E/FEN-TX-L3] (): bridge packetFilter list

   Packet Filter 1 - rejFddiat
      No port assignments
```

6 At the module prompt, enter **`bridge packetFilter assign`** to assign filters to port(s).

The Packet Filtering Language

You define packet filters using a *stack-oriented* language, which uses a LIFO (last in, first out) queue when the packet filter is running. The program places values (called *operands*) on the stack and tests them with various logical expressions (called *operators*), such as *and*, *or*, *equal*, and *not equal.* These expressions typically test the values of various fields in the received packet, which include MAC addresses, type fields, IP addresses, or any field within the first 64 bytes of any frame.

Principles for Writing a Custom Filter

Before you write a packet filter, understand these basic principles:

- How the Packet Filter Language Works
- What Can You Filter?
- Implementing Sequential Tests in a Packet Filter

A packet filter program is stored in a preprocessed format to minimize the space that is required by the packet filter definition. Comments are stripped. When assigned to a port, the packet filter is converted from the stored format to a run-time format to optimize the performance of the filter. Each module is limited to a maximum of 16 packet filter programs.

How the Packet Filter Language Works

A program in the packet filter language typically consists of a series of one or more instructions that results in the top of the stack containing a byte value after execution of the last instruction in the program. This top-of-stack byte value determines whether to forward or discard the packet.

In this stack-oriented language, instructions:

- *Push* operands onto the stack
- *Pop* the operands from the stack for comparison purposes
- *Push* the results back onto the stack

Therefore, with the exception of the push instructions, instructions (such as logical operators) locate their operands implicitly and do not require additional operand specifiers in the instruction stream.

Opcodes are the variables that are used to identify the type of operands and operators you are specifying in the packet filter instructions.

Procedure for Writing a Custom Filter

This section describes the process of writing a packet filter. Detailed examples are provided in "Long Custom Filter Example" later in this chapter.

You write the instructions for the packet filter using the following syntax:

```
<opcode>[.<size>] [<operand>...] [# <comment>]
```

The opcode descriptions are in "Packet Filter Opcodes" later in this chapter. Table 64 describes the supported operand sizes later in this chapter. The operand value is determined by what you are testing (for example, an address or a length).

Implicit operands for an instruction must be of the size expected by the instruction. Any mismatch in implicit operand size results in an error `operand size mismatch` *when you load the program into the system.*

When you write a packet filter, be sure that you use comments (preceded by #) to describe each step in the filter. This habit helps you to revise filters and enables others to understand and use the filters you create.

To write a packet filter, follow these basic steps:

1. Assign a unique, descriptive name to the filter using the `Name` opcode.
2. Specify what to test. For example, use the `pushField` opcode to select a field in the packet.
3. Specify what to compare to the value in step 2. For example, use the `pushLiteral` opcode to select a constant value.
4. Apply a logic operation to the values in steps 2 and 3. The operator you use depends on what comparison you want to make.

Table 64 describes the instructions and stacks of a packet filter.

Table 64 Packet Filter Instructions and Stacks — Descriptions and Guidelines

Element		**Descriptions and Guidelines**
Instructions		Each instruction in a packet filter definition must be on a separate line in the packet filter definition file.

	Instruction format	A typical instruction consists of an *opcode* followed by explicit *operands* and a *comment*. Although comments are optional, it is recommended that you use them throughout the packet filter to make it easier for yourself and others to administer the filters. Several opcodes include an explicit operand size specification.
		The general syntax of an instruction is:
		`<opcode>[.<size>] [<operand>...] [# <comment>]`
		Example:
		`pushliteral.l  0xffffff00  #load the type field mask`
		Use any combination of uppercase and lowercase letters for the opcode and size.
		The contents of a line following the first # outside a quoted string are ignored, so use the # to begin your comments. Comments are not stored in the system; they are useful when the filter is created and saved externally.

	Operand sizes	The following operand sizes are supported: ■ 1 byte = .b ■ 2 bytes = .w ■ 4 bytes = .l ■ 6 bytes = .a (Included primarily for use with 48-bit, IEEE, globally assigned MAC addresses)
	Maximum length	The maximum length for a filter definition is 4096 bytes.
Stack		The packet filter language uses a *stack* to store the operands that will be used by an instruction and the results of the instruction. Operands are popped from the stack as required by the instructions. An instruction using two or more operands takes the first operand from the top of the stack, with subsequent operands taken in order from succeeding levels of the stack. The stack is a maximum of 64 bytes long, with space within the stack allocated in multiples of 4 bytes. Thus you can have a maximum of 16 operands on the stack. The address size operand .a consumes 8 bytes on the stack, decreasing the maximum number of operands on the stack for a 48-bit address.

The Ethernet and FDDI packet fields in Figure 24 are used as *operands* in the packet filter. The two simplest operands are described in Table 65.

Table 65 Two Packet Filter Operands

Operand	Description	Opcode
packet field	A field in the packet that can reside at any offset. The size of the field can be 1, 2, 4, or 6 bytes. Typically, you only specify a 6-byte field when you want the filter to examine a 48-bit address.	pushField
constant	A literal value to which you are comparing a packet field. As with a field, a constant can be 1, 2, 4, or 6 bytes long.	pushLiteral

Packet Filter Opcodes

Opcodes are instructions used in packet filter definitions. The available opcodes are described in Table 66.

Table 66 Packet Filtering Opcodes

Opcode	Memory Requirements	Description
name "<name>"	2 + *n* bytes, where *n* is the length of the <name>	Assigns a user-defined <name> to the packet filter. The name may be any sequence of ASCII characters other than quotation marks. The name is limited to 32 characters. You can include only a single name statement in each packet filter program.
pushField.size <offset>	3 bytes	Pushes a field from the target packet onto the stack. Packet data starting at <offset> is copied onto the stack. The most significant byte of the field is the byte at the specified offset. The size field of the instruction determines the number of bytes pushed. The pushField instruction provides direct access to any 1, 2, 4, or 6 byte (.b, .w, .l, or .a) field contained within the first 64 bytes of the target packet. Specify the offset as an octal, decimal, or hexadecimal number. ■ Precede an octal number by a "0". ■ Precede a hexadecimal number by either "0x" or "0X". ■ Use either upper or lower case letters for the hexadecimal digits "a" through "f".

Table 66 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description
pushLiteral.size <value>	1 (.b) 2 (.w) 4 (.l) 6 (.a) bytes depending on the size of <value> plus 1 byte for a total of 2, 3, 5, or 7 bytes	Pushes a literal constant <value> onto the stack. The most significant byte of the <value> is the first byte of the literal. Bytes are copied directly from the operand onto the stack. The size field of the instruction determines number of bytes pushed. Specify the value as either an octal, decimal, or hexadecimal number. ■ Precede an octal number by a "0". ■ Precede a hexadecimal number by either "0x" or "0X". ■ Use either upper or lower case letters for the hexadecimal digits "a" through "f".
pushTop	1 byte	Pushes the current top of the stack onto the stack (that is, it reads the top of the stack and pushes the value onto the stack, which effectively duplicates the item currently on top of the stack). The size of the contents of the stack determines the size of the push. Use `pushTop` for each additional comparison you intend to make with the current top of the stack. The `pushTop` instruction makes a copy of the field more efficiently than if you use a second `pushField` instruction. If you are writing a filter that is going to check the same offset more than once, such as checking the Ethernet type field to filter multiple protocols, use the following guidelines. Assume that you want to filter DEC LAT, IP, and ARP traffic on a port.

Table 66 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description
pushLiteral.size <value>	1 (.b) 2 (.w) 4 (.l) 6 (.a) bytes depending on the size of <value> plus 1 byte for a total of 2, 3, 5, or 7 bytes	Pushes a literal constant <value> onto the stack. The most significant byte of the <value> is the first byte of the literal. Bytes are copied directly from the operand onto the stack. The size field of the instruction determines number of bytes pushed. Specify the value as either an octal, decimal, or hexadecimal number. ■ Precede an octal number by a "0". ■ Precede a hexadecimal number by either "0x" or "0X". ■ Use either upper or lower case letters for the hexadecimal digits "a" through "f".
pushTop	1 byte	Pushes the current top of the stack onto the stack (that is, it reads the top of the stack and pushes the value onto the stack, which effectively duplicates the item currently on top of the stack). The size of the contents of the stack determines the size of the push. Use `pushTop` for each additional comparison you intend to make with the current top of the stack. The `pushTop` instruction makes a copy of the field more efficiently than if you use a second `pushField` instruction. If you are writing a filter that is going to check the same offset more than once, such as checking the Ethernet type field to filter multiple protocols, use the following guidelines. Assume that you want to filter DEC LAT, IP, and ARP traffic on a port.

Table 66 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description

pushTop (continued)	1 byte	Rather than use multiple `pushField .w 12` commands to look at the 12th offset where the Ethernet type field resides, use multiple `pushTop` commands, as shown here:
		
		Original Filter:
		
		`pushField.w 12`
		`pushLiteral.w 0x6004`
		`eq`
		`reject`
		`pushField.w 12`
		`pushLiteral.w 0x0800`
		`eq`
		`reject`
		`pushField.w 12`
		`pushLiteral.w 0x0806`
		`ne`
		
		Shortened Filter:
		
		`PushField.w 12`
		`pushTop`
		`pushTop`
		`pushLiteral.w 0x6004`
		`eq`
		`reject`
		`pushLiteral.w 0x0800`
		`eq`
		`reject`
		`pushLiteral.w 0x0806`
		`ne`

pushSPGM	1 byte	Pushes the source port group mask (SPGM) onto the top of the stack. The SPGM is a bitmap representing the groups to which the source port of a packet belongs. This instruction pushes 4 bytes on to the stack. Each port group mask is represented by a single bit in the SPGM bitmap. Port group masks are assigned to the bitmap in sequence, starting with port group mask 1 as the least significant bit through port group mask 32 as the most significant bit. Use `pushSPGM` to filter by port groups. See "Using Port Groups in Custom Packet Filters" for more information.

Table 66 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description
pushDPGM	1 byte	Pushes the destination port group mask (DPGM) onto the top of the stack. The DPGM is a bitmap representing the groups to which the destination port of a packet belongs. Pushes 4 bytes on to the stack. Each port group mask is represented by a single bit in the DPGM bitmap. Port group masks are assigned to the bitmap in sequence, starting with port group mask 1 as the least significant bit through port group mask 32 as the most significant bit. Use `pushDPGM` to filter by port groups. See "Using Port Groups in Custom Packet Filters" for more information.
eq (equal)	1 byte	Pops two values from the stack and compares them. If they are equal, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determines the size of the operands.
ne (not equal)	1 byte	Pops two values from the stack and compares them. If they are not equal, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The size of the operands is determined by the contents of the stack.
lt (less than)	1 byte	Pops two values from the stack and performs an unsigned comparison. If the first is less than the second, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determine the size of the operands.
le (less than or equal to)	1 byte	Pops two values from the stack and performs an unsigned comparison. If the first is less than or equal to the second, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determine the size of the operands.
gt (greater than)	1 byte	Pops two values from the stack and performs an unsigned comparison. If the first is greater than the second, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determine size of the operands.

Table 66 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description
ge (greater than or equal to)	1 byte	Pops two values from the stack and performs an unsigned comparison. If the first is greater than or equal to the second, a byte containing the non-zero value is pushed onto the stack; otherwise, a byte containing 0 is pushed. The contents of the stack determine the size of the operands.
and (bit-wise AND)	1 byte	Pops two values from the stack and pushes the bit-wise *AND* of these values back onto the stack. The contents of the stack determine the size of the operands and the result. This is a bit-wise operator. Each bit of the operands is logically compared to produce the resulting bit
or (bit-wise OR)	1 byte	Pops two values from the stack and pushes the bit-wise *OR* of these values back onto the stack. The contents of the stack determine the operand size and the result. This is a bit-wise operator. Each bit of the operands is logically compared to produce the resulting bit
xor (bit-wise exclusive-OR)	1 byte	Pops two values from the stack and pushes the bit-wise exclusive-*OR* of these values back onto the stack. The contents of the stack determines the operand size and the result. This is a bit-wise operator. Each bit of the operands is logically compared to produce the resulting bit
not	1 byte	Pops a byte from the stack; if its value is non-zero, a byte containing 0 is pushed back onto the stack. Otherwise, a byte containing the value is pushed back onto the stack.
accept	1 byte	Conditionally accepts the packet that is being examined. Pops a byte from the stack. If its value is non-zero, the packet is accepted and evaluation of the filter ends immediately; otherwise, filter evaluation continues with the next instruction. Use `accept` with `and` and `or` operators when you have sequential tests and you would like the filter to accept a packet before the entire expression has been evaluated. Using `accept` can significantly improve the performance of certain types of filters. See "Implementing Sequential Tests in a Packet Filter" elsewhere in the chapter for more information.

Table 66 Packet Filtering Opcodes (continued)

Opcode	Memory Requirements	Description
reject	1 byte	Conditionally rejects the packet being examined. Pops a byte from the stack. If its value is non-zero, the packet is rejected and filter evaluation ends immediately; otherwise, the filter evaluation continues with the next instruction. Use `reject` with `and` and `or` operators when you have sequential tests and you would like the filter to reject a packet before the entire expression has been evaluated. Using `reject` can significantly improve the performance of certain types of filters. See "Implementing Sequential Tests in a Packet Filter" earlier in the chapter for more information.
shiftl (shift left)	1 byte	Pops two values from the stack and shifts the first operand left by the number of bits specified by the second operand. Bits shifted out of the left side of the operand are discarded, and zeros are shifted in from the right. The resulting value is pushed back onto the stack. The contents of the top of the stack determines the size of the first operand and the size of the result. The second operand is always 1 byte and only the low 5 bits of the byte are used as the shift count.
shiftr (shift right)	1 byte	Pops two values from the stack and shifts the first operand right by the number of bits specified by the second operand. Bits shifted out of the right side of the operand are discarded, and zeros are shifted in from the left. The resulting value is pushed back onto the stack. The contents of the top of the stack determines the size of the first operand and the size of the result. The second operand is always 1 byte and only the low 5 bits of the byte are used as the shift count.

Implementing Sequential Tests in a Packet Filter

Filter language expressions are normally evaluated to completion — a packet is accepted if the value remaining on the top of the stack is nonzero. Frequently, however, a single test is insufficient to filter packets effectively. When more tests are warranted, you want to accept a packet that satisfies one of two cases:

- At least one criterion specified in two or more tests (that is, ORs the results of the tests)

  OR

- All criteria specified in two or more tests (that is, ANDs the results of the tests)

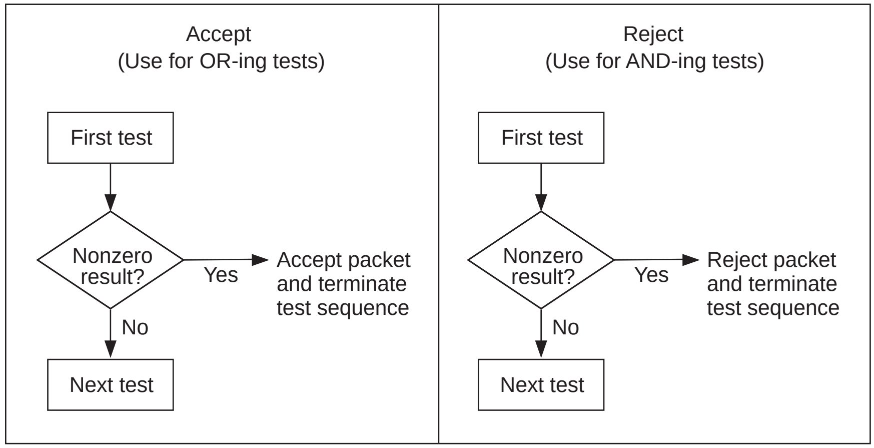
The *accept* and *reject* instructions are used to implement sequential tests, as shown in Figure 26.

In order to optimize a filter's performance, it is best to exit a filter as early as possible. If you wait until the last instruction to make the forward or filter decision, more processing is needed.

The accept and reject criteria allow you to exit a filter early. When using these instructions, construct the packet filter so that tests that apply to the majority of the network traffic are performed first. This ensures that the filter is exited after the first instruction for the majority of packets. Only a small number of packets will require additional tests.

For example, assume you want to create a filter that checks for particular IPX attributes that you want to filter, but most of the traffic on your network is IP traffic. In this case, it would be best to first check each packet to see if it is a IP frame. If it is, you could accept the packet immediately. Now only the smaller number of packets that contain IPX information would be subjected to additional tests.

Figure 26 Accept and Reject Instructions



The following example shows the use of both accept and reject in a packet filter. This packet filter was created for a network that is running both Phase I and Phase II AppleTalk software. The goal of the filter is to eliminate the AppleTalk traffic.

```
Name       “Filter AppleTalk datagrams”
pushField.w       12                          # Get the type field.
pushTop                                       # Make a copy.
pushLiteral       0x809b                      # EtherTalk Phase I type.
eq                                            # Test if the packet type is
                                              # equal to the AppleTalk type,
reject                                        # reject the packet and end.
                                              # Otherwise,
pushLiteral.w     0x5dc                       # Largest 802.3 packet size
lt                                            # If this value is less than the
                                              # value in the packet’s
                                              # type/length field, then this
                                              # is an Ethernet frame, so
accept                                        # accept the packet if it is not
                                              # 802.3, otherwise…
pushField.a       16                          # get the SNAP OUI and Ethertype
pushLiteral.a     0x03080007809b              # value to compare.
ne                                            # If not equal, then forward the
                                              # packet, otherwise drop it.
```

Common Syntax Errors

When you leave the Administration Console's built-in editor or load a packet filter definition from across the network, the software examines the definition for syntax errors. Table 67 lists syntax errors and their causes.

Table 67 Common Syntax Errors

Syntax Error	Description
Opcode not found OR Unknown opcode	An opcode was expected on the line and was not found. The opcode must be one of those described in "Packet Filter Opcodes" later in this chapter and must include the size, if any. The opcode and size must be separated by a single period (.) with no intervening spaces. Any mix of uppercase and lowercase characters is permitted.
Operands are not the same size	The opcode requires two operands of the same size. The top two operands on the stack are of different sizes.
Stack underflow	The opcode requires one or more operands. An insufficient number of operands are currently on the stack.
Stack overflow	The opcode pushes an operand on the stack. The stack does not have sufficient room for the operand.
No result found on top of stack	The program must end with a byte operand on the top of the stack. After the last instruction in the program is executed, the stack is either empty or contains an operand other than a byte.
Extra characters on line	The source line contains extraneous characters that are not part of the instruction and are not preceded by a comment character (#).
Expected a byte operand	The opcode requires a byte operand as one of its parameters. The operand is of a size other than a byte.
Offset not found	The opcode requires an offset to be specified. None was found on the line.
Literal not found	The opcode requires a literal value to be specified. None was found on the line.
String not found	The opcode requires a quoted string to be specified. None was found on the line.

Table 67 Common Syntax Errors (continued)

Syntax Error	Description
Invalid characters in number	The number specified as an offset or literal is improperly formatted. Possible causes are 1) lack of white space setting off the number, and 2) invalid characters in the number. Note: The radix of the number is determined by the first 1 or 2 characters of the number: ■ A number with a leading "0x" or "0X" is treated as hexadecimal. ■ All other numbers with a leading 0 are treated as octal. ■ All other numbers are treated as decimal.
Number is too large	The number that is specified as an offset or literal is too large. An offset is limited to 1518 minus the size of the operand. For example, the offset for pushField.b can be no more than 1517, and the offset for pushField.w no more than 1516. A literal value is limited to the number of bytes in the operand size (1, 2, 4, or 6).
Missing open quote on string	The string specified does not have a starting quotation mark (").
String is too long	The string specified is too long. Strings are limited to 32 characters exclusive of the opening and closing quotation marks.
Missing close quote on string	The string specified does not have an ending quotation mark (").
Multiple name statements in program	More than one name statement was found in the program. Only a single name statement is allowed.
Program too large	The program exceeds the maximum size allowed. The causes of this error include a source definition exceeding 4096 bytes, a stored format exceeding 254 bytes, or a run-time format exceeding 2048 bytes. All of these boundary conditions are checked when the filter is loaded.
Too many errors – compilation aborted	The program contains an excessive number of errors. No further syntax errors will be reported. The program stops compiling when this condition occurs.

Custom Packet Filter Examples

The following examples of packet filters, which were built using the packet filter language, start with basic concepts.

Destination Address Filter

This filter operates on the destination address field of a frame. It allows packets to be forwarded that are destined for stations with an Organizationally Unique Identifier (OUI) of 08-00-02. To customize this filter to another OUI value, change the literal value loaded in the last pushLiteral.l instruction. The OUI must be padded with an additional 00 to fill out the literal to 4 bytes.

```
name            “Forward to 08-00-02”
pushField.l     0               # Get first 4 bytes of
                                # destination address.
pushLiteral.l   0xffffff00      # Set up mask to isolate first
                                # 3 bytes.
and                             # Top of stack now has OUI
pushLiteral.l   0x08000200      # Load OUI value.
eq                              # Check for match.
```

Source Address Filter

This filter operates on the source address field of a frame. It allows packets to be forwarded that are from stations with an OUI of 08-00-02. To customize this filter to another OUI value, change the literal value loaded in the last pushLiteral.l instruction. The OUI must be padded with an additional 00 to fill out the literal to 4 bytes.

```
name            “Forward from 08-00-02”
pushField.l     6               # Get first 4 bytes of source
                                # address.
pushLiteral.l   0xffffff00      # Set up mask to isolate first
                                # 3 bytes.
and                             # Top of stack now has OUI
pushLiteral.l   0x08000200      # Load OUI value.
eq                              # Check for match.
```

Length Filter

This filter operates on the length field of a frame. It allows packets to be forwarded that are less than 400 bytes in length. To customize this filter to another length value, change the literal value loaded in the pushLiteral.w instruction.

```
name            “Forward < 400”
pushField.w     12              # Get length field.
pushLiteral.w   400             # Load length limit.
lt                              # Check for frame length <
                                # limit.
```

Type Filter

This filter operates on the type field of a frame. It allows packets to be forwarded that are IP frames. To customize this filter to another type value, change the literal value loaded in the pushLiteral.w instruction.

```
name            "Forward IP frames"
pushField.w     12              # Get type field.
pushLiteral.w   0x0800          # Load IP type value.
eq                              # Check for match.
```

Ethernet Type IPX and Multicast Filter

This filter *rejects* frames that have either a Novell IPX Ethernet type field (8134 hex) or a multicast destination address.

```
name            "Type > 900 or Multicast"
pushField.w     12              # Get type field.
pushLiteral.w   0x900           # Push type value to test
                                # against.
gt                              # Is type field > 900 (hex)?
reject                          # If yes: reject frame (done).
pushLiteral.b   0x01            # Multicast bit is low-order
pushField.b     0               # bit
and                             # Get 1 st byte of destination
not                             # Isolate multicast bit
                                # Top of stack 1 to accept,
                                # 0 to reject
```

Multiple Destination Address Filter

This filter operates on the destination address field of a frame. It allows packets to be forwarded that are destined for one of four different stations. To tailor this filter to other destinations, change the literal values.

```
name            "Forward to four stations"
pushField.a     0               # Get destination address.
pushTop                         # Make 3 copies of address.
pushTop                         #
pushTop                         #
pushLiteral.a   0x367002010203  # Load allowed destination
                                # address.
eq                              # Check for match.
accept                          # Forward if valid address.
pushLiteral.a   0x468462236526  # Load allowed destination
                                # address.
eq                              # Check for match.
accept                          # Forward if valid address.
pushLiteral.a   0x347872927352  # Load allowed destination
                                # address.
eq                              # Check for match.
accept                          # Forward if valid address.
pushLiteral.a   0x080239572897  # Load allowed destination
                                # address.
eq                              # Check for match.
```

Source Address and Type Filter

This filter operates on the source address and type fields of a frame. It allows XNS packets to be forwarded that are from stations with an OUI of 08-00-02. To customize this filter to another OUI value, change the literal value loaded in the last pushLiteral.l instruction. You must pad the OUI with an additional 00 to fill out the literal to 4 bytes. To customize this filter to another type value, change the literal value loaded into the pushLiteral.w instruction.

```
name            “XNS from 08-00-02”
pushField.w     12              # Get type field.
pushLiteral.w   0x0600          # Load type value.
ne                              # Check for mis-match.
reject                          # Toss any non-XNS frames.
pushLiteral.l   0xffffff00      # Set up mask to isolate first 3
                                # bytes.
pushField.l     6               # Get first 4 bytes of source
                                # address.
and                             # Top of stack now has OUI.
pushLiteral.l   0x09000200      # Load OUI value.
eq                              # Check for match.
```

Accept XNS or IP Filter

This filter operates on the type field of a frame. It allows packets to be forwarded that are XNS or IP frame. The pushTop instruction makes a copy of the type field.

```
name            “Forward IP or XNS”
pushField.w     12              # Get type field.
pushTop                         # Push copy of type.
pushLiteral.w   0x0800          # Load IP type value.
eq                              # Check for match.
pushLiteral.w   0x0600          # Load XNS type value.
eq                              # Check for match.
```

XNS Routing Filter

This filter operates on the type and data fields of a frame. It discards all XNS routing packets.

```
name            “Drop XNS Routing”
pushField.w     12              # Get type field.
pushLiteral.w   0x0600          # Load XNS type value.
ne                              # Check for non-XNS packet.
accept                          # Forward if non-XNS packet.
pushLiteral.b   0x01            # Load XNS routing type.
pushField.b     19              # Get XNS type.
ne                              # Check for non-XNS routing
                                # packet.
```

Port Group Filter

See "Using Port Groups in Custom Packet Filters" for a port group filter example.

Limits to Filter Size

A packet filter program is stored in a preprocessed format to minimize the space that is required by the packet filter definition. Comments are stripped. When assigned to a port, the packet filter is converted from the stored format to a run-time format to optimize the performance of the filter. Each module is limited to a maximum of 16 packet filter programs.

The maximum length of a packet filter definition is 4096 bytes.

Storage Rules for Preprocessed Packet Filters

Each module provides a maximum of 2048 bytes of nonvolatile storage for *preprocessed* packet filter programs. In the preprocessed stored format:

- A single packet filter program is limited to 254 bytes.
- Each instruction in the packet filter program requires 1 byte for the opcode and size, plus additional bytes for any explicit operands.
- Module overhead is 22 bytes, plus a per-packet-filter overhead of 13 bytes. For example, assume a packet filter program requires 200 bytes for storing the instructions in the program. If this packet filter is the only one loaded, the nonvolatile memory required is 22 bytes (for module overhead) plus 13 bytes (for packet filter overhead) plus 200 bytes (for the program itself) — a total of 235 bytes.

Run-time Storage of Packet Filters

For *run-time* storage of packet filter programs, each module provides a maximum of 8192 bytes. There is no explicit system or per-packet-filter overhead; however, performance considerations can result in unused areas of the run-time storage.

The run-time format is approximately eight times the size of the stored format. Thus a 200-byte packet filter program in stored format expands to approximately 1600 bytes in the run-time format. A single packet filter program cannot exceed 2048 bytes in the run-time format.

Using Port Groups in Custom Packet Filters

You can use a port group (a list of module ports) as filtering criteria in a packet filter.

A packet filter uses the group to make filtering decisions by accessing the group's source port group mask and destination port group mask. In the mask, 32 bits indicate to which of 32 possible groups a port belongs. For example, setting mask bit number 7 assigns the port to group number 7.

You reference these group masks using the opcodes SPGM (source port group mask), and DPGM (destination port group mask). What follows is an example of using port groups in packet filters.

Port Group Packet Filter Example

In this example, packets are not forwarded to ports in groups 3 and 8.

```
Name    “Discard Groups 3 and 8”
pushSPGM                              # Get source port group mask.
pushLiteral.l    0x0084               # Select bits 3 and 8.
and                                   # If port group bits 3 & 8 are common
                                      # with SPGM, then non-zero value is
                                      # pushed onto stack.
pushLiteral.l    0                    # Push zero.
eq                                    # Only if SPGM is not in port groups
                                      # corresponding to bits 3 & 8, then
                                      # packet is forwarded.
```

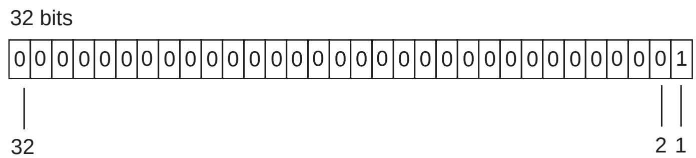
Port Group Filter Operation

When an address is learned on a port, the address and the port number the packet was received on are inserted into the bridge address table and a bit mask that is associated with the address that denotes the group membership is inserted into the port group mask table.

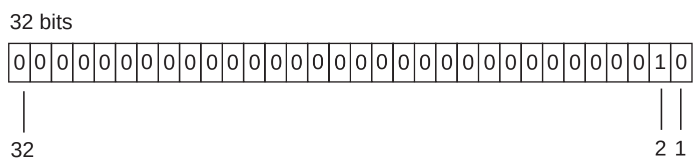
The bridge address table stores each SA/DA MAC address with the port number. The port group masks are stored in a smaller table associating port numbers to port group masks.

For example, assume you defined port group 1 with port 3 being a member and port group 2 with port 5 being a member.

If MAC address 00-80-3e-12-34-56 is learned on port 3 and port 3 belongs to port group 1, it has a port group bit mask for port group 1 inserted into the port group mask table that is associated with the MAC address in the bridge address table. The mask is 32 bits long and contains:



If MAC address 00-80-3e-aa-aa-aa is learned on port 5 and port 5 belongs to port group 2, it has a port group bit mask for port group 2 inserted into the port group mask table that is associated with the MAC address in the bridge address table. The mask is 32 bits long and contains:



When you use the source or destination port group mask (SPGM and DPGM) commands in your filter, you are referencing the port mask of the source port and the port mask of the destination port, respectively. You can use these commands to verify if the source and destination addresses of the packets are members of the same port group to implement your filtering algorithm.

A frame is received (unicast/multicast/broadcast) on the source port. The source port group mask (SPGM) is found in the table of port group masks, using the received port as the index. The destination port group mask (DPGM) is found after the bridge determines whether the port is to be forwarded (known DA unicast) or flooded (unknown DA unicast, or multicast, or broadcast).

For forwarded frames, the single SPGM and DPGM are used in the packet filter.

For flooded frames, each pair of SPGM and DPGM are individually processed. The filter is repeated for each pair of Source and Destination ports.

For example, port 1 has a packet filter using the DPGM assigned to port 1's rxAll path and a broadcast frame is received on port 1. The bridge determines that the frame will be flooded to the VLAN ports 2-5. The filter is processed 4 times:

1. Once for the RX port 1 - TX port 2 pair
2. Once for the RX port 1 - TX port 3 pair
3. Once for the RX port 1 - TX port 4 pair
4. Once for the RX port 1 - TX port 5 pair

Use the Standard Port Group filter to contain broadcast, multicast, and flooded frames:

```
pushSPGM
pushDPGM
and
```

If ports 1-3 are in port group 1, 4-5 are in port group 2, and the rxAll path filter is applied to 1-5, then the appropriate filtering restricts the flooding to the corresponding port group.

Table 68 and Table 69 show how each port pair filters or does not filter a broadcast frame that is received on port 1 and destined for ports 2,3,4,5:

Table 68 Port Group Mask Table

Port	Mask
1	0x00000001
2	0x00000001
3	0x00000001
4	0x00000002

Table 69 Filter Processing for Frame Flooded from Port 1 to Ports 2-5

Port Pair	SPGM	DPGM	Filter Result
1 and 2	0x00000001	0x00000001	0x00000001 - ACCEPTED.
1 and 3	0x00000001	0x00000001	0x00000001 - ACCEPTED.
1 and 4	0x00000001	0x00000002	0x00000000 - FILTERED.
1 and 5	0x00000001	0x00000002	0x00000000 - FILTERED.

The result is that the frame is flooded to ports 2,3, and the frame is filtered from ports 4,5.

Port Group Management and Control Functions

This section describes the management and control functions that you use to define port groups.

Defining Port Groups

You can configure port groups from the `bridge packetFilter portGroup` menu of the Administration Console.

This section briefly discusses the control and management functions that are implemented in the module for port groups.

You need to assign the port groups before you can assign packet filters to the port groups.

Important Considerations

- **Creating a new group** — When you create a new port group, an unused port group must be available. A port group is limited to the number of ports on a module.
- **Listing groups** — You can list the port groups currently defined on the module. The group id, group name (if any), group mask, and the slots where the group is loaded are displayed.
- **Displaying groups** — The display of a port group shows the group id, the name of the group, and all the addresses or ports included in that group.
- **Deleting groups** — When you delete port groups from a module, those groups are no longer available for use in packet filters.
- **Adding Ports to Groups** — When you add ports to an existing group, you can either enter the ports at the prompts or import them from a file. At least one port group must exist before you can add ports. The same port may be in multiple port groups.
- **Port group size** — The maximum number of ports that a port group can contain is 24, which is the maximum number of ports on a system.
- **Removing ports from a group** — At least one group must exist before you can remove a port.

- **Loading groups** — The Administration Console has no explicit menu item for loading port groups that are defined in a file on a remote host. However, you can *load* groups by creating a script on a remote host (which includes your port group) and then running that script on your Administration Console host.

The following example shows a script that builds two port groups: one named Mktg and the other named Sales:

```
bridge packetFilter portGroup create
15
Mktg
1,2,3
bridge packetFilter portGroup create
32
Sales
5,6
```

When you run the script, your groups are automatically created and stored on the system.

Long Custom Filter Example

The following solution shows a complex packet filter built from three simple packet filters. Each of the shorter, simpler packet filters can be used on its own to accomplish its own task. Combined, these filters create a solution for a larger filtering problem.

Filtering Problem

Your network contains market data feed servers that receive time-critical financial data needed for trading floor applications. At the center of the trading floor networks is a system that is being used to switch Ethernet traffic and to concentrate the market data feed servers onto the FDDI departmental backbone.

The difficulty is that the market data feed servers transmit data to users with broadcast packets that are forwarded to all stations on all segments attached to the system. Not all of the segments attached to the system have stations that require these broadcast updates. To optimize the performance of these Ethernet segments, you need to filter the broadcasts.

Packet Filter Solution

The solution described here is to create a highly sophisticated packet filter that prevents only the broadcast packets from the market data servers from being forwarded onto the segments that are not part of an active trading floor.

Before you write the packet filter, it is important to understand the functions that the filter must provide. The broadcast packets that are transmitted by the servers are based on either TCP/IP or XNS protocol. In both cases, the broadcast packets have socket values that are greater than 0x076c and less than 0x0898. The socket value is located 24 bytes into the packet in IP datagrams and 30 bytes into the packet in XNS datagrams.

You can use this information to create pseudocode that simplifies the process of writing the actual filter. It helps to first write the pseudocode in outline form, as shown here:

1. Determine if the packet has a broadcast address.
2. Determine if the packet is an XNS datagram.
3. Examine socket values and discard the packet if:
   - The socket value is greater than or equal to 0x76c

     AND
   - The socket value is less than 0x898
4. Determine if the packet is an IP datagram.
5. If so, then examine socket values and discard the packet if:
   - The socket value is greater than or equal to 0x76c

     AND
   - The socket value is less than 0x898
6. End the filter.

The pseudocode translates into the following complex packet filter:

```
Name            "IP XNS ticker bcast filter"
                              # Assign this filter in the multicast path
                              # of a port only--this is very important.
                              #
                              # XNS FILTERING SECTION
                              #
pushField.a     0             # Apply
pushLiteral.a   0xffffffffffff# filter
ne                            # only on broadcast traffic
accept                        #
pushField.w     12            # Get the type field of the packet and
                              # place it on top of the stack.
pushLiteral.w   0x0600        # Put the type value for XNS on top of
                              # the stack.
eq                            # If the two values on the top of the
                              # stack are equal, then return a non-zero
                              # value.
pushLiteral.w   0x76c         # Put the lowest socket value on top of
                              # the stack.
pushField.w     30            # Put the value of the socket from the
                              # packet on top of the stack.
ge                            # Compare if the value of the socket is
                              # greater than or equal to lower bound.
pushLiteral.w   0x0898        # Put the highest socket value on top of
                              # the stack.
pushField.w     30            # Put the value of the socket from the
                              # packet on top of the stack.
lt                            # Compare if the value of the socket is
                              # less than the upper bound
and                           # "and" together with "ge" and "lt" test
                              # to determine if the socket value is
                              # "within" the range. If it is, place a
                              # "one" on the stack
and                           # Compare if XNS & in range
                              #
                              # IP FILTERING SECTION
                              #
pushField.w     12            # Get the type field of the packet and
                              # place it on top of the stack.
pushLiteral.w   0x0800        # Put the type value for IP on top of
                              # the stack.
eq                            # If the two values on the top of the
                              # stack are equal, then return a non-zero
                              # value.
pushLiteral.w   0x76c         # Put the lowest socket value on top of
                              # the stack (1900).
pushField.w     24            # Put the value of the socket from the
                              # packet on top of the stack.
ge                            # Compare if the value of the socket is
                              # greater than or equal to lower bound.
pushLiteral.w   0x0898        # Put the highest socket value on top of
                              # the stack (2200).
pushField.w     24            # Put the value of the socket from the
                              # packet on top of the stack.
lt                            # Compare if the value of the socket is
                              # less than the upper bound
and                           # "and" together with "ge" and "lt".
                              # Test to determine if the socket value is
                              # "within" the range. If it is in range,
                              # place a "one" will on the stack.
and                           # Compare if IP and in range.
or                            # Determine if the type field is either
                              # XNS or IP.
not                           # Discard if (IP & in range) and (XNS & in
                              # range).
```

The rest of this section concentrates on the parts of the complex filter, showing you how to translate the pseudocode's requirements into filter language. The large filter is broken down into subsets to show how you can create small filters that perform one or two tasks, and then combine them for more sophisticated filtering. Table 70 describes how the purpose of each pseudocode step is accomplished in the small series of packet filters.

Table 70 Pseudocode Requirements Mapped to the Packet Filter

Step	Accomplished through
1	The path to which you assign the packet filter. For administrative purposes, this path is specified in the first two comment lines in the filter definition. The filter must be assigned to a multicast path to filter packets that have broadcast addresses.
2	Packet Filter One — Forwarding XNS packets
3	Packet Filter Two — Looking for specified socket range
4 & 5	Combining a Subset of Filters — Forwarding IP packets within specified socket range

Packet Filter One

This filter is designed to leave a non-zero value on the stack for XNS broadcast packets. If used alone, this filter accepts the very packets we are trying to filter. The reason for doing this will become clear when the filter is combined later in this section.

These steps show how to create this filter.

1 Name the filter:

```
Name    “Forward only XNS packets”
```

It is important to distinguish the function of each filter when it is loaded onto a system that has more than one filter stored in memory. Naming is also useful for archiving filters on a remote system so that the filters can be saved and loaded on one or more systems.

2 Enter executable instruction #1:

```
pushField.a    0
# Clear the stack
```

3 Enter executable instruction #2:

```
pushField.a    0xffffffffffff
# Put the broadcast address on the top of the stack
```

4 Enter executable instruction #3:

```
ne
# not 0xffffffffffff
```

5 Enter executable instruction #4:

```
accept
# accept packet and go no further
```

This accepts all non-broadcast packets.

6 Enter executable instruction #5:

```
pushField.w    12
# Get the type field of the packet and
# place it on top of the stack.
```

7 Enter executable instruction #6:

```
pushLiteral.w    0x0600
# Put the type value for XNS on top
# of the stack.
```

8 Enter executable instruction #7:

```
eq
# If the two values on the top of the stack are equal,
# then return a non-zero value.
```

This returns non-zero for XNS broadcast frames.

Packet Filter Two

This filter is designed to accept packets within the socket range of 0x76c and 0x898. When combined with Filter One above, it forwards XNS packets. Follow these steps to create this filter.

1 Name the filter:

```
Name    "Socket range filter"
```

2 Enter executable instruction #1:

```
pushLiteral.w    0x76c
# Put the lowest socket value on top
# of the stack.
```

3 Enter executable instruction #2:

```
pushField.w    30
# Put the value of the socket from the
# packet on top of the stack.
```

4 Enter executable instruction #3:

```
ge
# Compare if the value of the socket is greater than
# or equal to the lower bound.
```

5 Enter executable instruction #4:

```
pushLiteral.w    0x0898
# Put the highest socket value on
# top of the stack.
```

6 Enter executable instruction #5:

```
pushField.w    30
# Put the value of the socket from the
# packet on top of the stack.
```

7 Enter executable instruction #6:

```
lt
# Compare if the value of the socket is less than the
# upper bound.
```

8 Enter executable instruction #7:

```
and
# “and” together with “ge” and “lt” test to determine
# if the socket value is “within” the range. If it is,
# place a non-zero value on the stack.
```

Combining a Subset of the Filters

The next filter places a non-zero value on the stack for IP packets with a socket range of 0x76c (1900) and 0x898 (2200). The filter combines packet filters one and two, modifying them for IP. These steps show how to create this filter.

1 Name the filter:

```
name    “Only IP pkts w/in socket range”
```

2 Perform steps 6 through 8 as described earlier in “Packet Filter One” except give the pushLiteral instruction (in step 7) a value of 0x0800 for IP.

3 Perform steps 2 through 8 as described earlier in “Packet Filter Two” except the socket value for IP (in steps 3 and 6) is located 24 bytes into the packet (instead of 30 as for XNS).

4 Add an *and* statement to compare the results of step 2 with the results of step 3:

```
and
# Compare if IP and in range.
```

This combination looks like this:

```
Name    “Only IP pkts w/in socket range”
pushField.w     12              # Get the type field of the packet and
                                # place it on top of the stack.
pushLiteral.w   0x0800          # Put the type value for IP on top of
                                # the stack.
eq                              # If the two values on the top of the
                                # stack are equal, then return a non-zero
                                # value.
pushLiteral.w   0x76c           # Put the lowest socket value on top of
                                # the stack (1900).
pushField.w     24              # Put the value of the socket from the
                                # packet on top of the stack.
ge                              # Compare if the value of the socket is
                                # greater than or equal to the lower bound
pushLiteral.w   0x0898          # Put the highest socket value on top of
                                # the stack (2200).
pushField.w     24              # Put the value of the socket from the
                                # packet on top of the stack.
lt                              # Compare if the value of the socket is
                                # less than the upper bound.
and                             # “and” together with “ge” and “lt” test
                                # to determine if the socket value is
                                # “within” the range. If it is in range,
                                # place a “one” will on the stack.
and                             # Compare if IP and in range.
```

Combining All the Filters

Together, the packet filters work to perform the solution to the problem: filtering the broadcast packets from the market data servers. These steps show how to create this filter:

1 Name the filter:

```
name    “Discard XNS & IP broadcast pkts w/in socket range”
```

2 Perform steps 2 through 8 as described earlier in “Packet Filter One.”

3 Perform steps 2 through 8 as described earlier in “Packet Filter Two.”

4 Add an *and* statement to compare the results of step 2 and the results of step 3:

```
and
# compare if XNS & in range
```

5 Perform steps 2 through 4 as described earlier in “Combining a Subset of the Filters.”

6 Add an *or* statement:

```
or
# determine if the type field is either XNS or IP
```

7 Add a *not* statement to discard any matching packets:

```
not
# discard if (IP & in range) or (XNS & in range)
```

The complete packet filter discards IP and XNS packets that are within the specified range.

Optimizing the Filter with Accept and Reject Commands

The following combination filter performs the same function but uses the `accept`, `reject`, and `pushTop` commands to exit the filter as soon as possible to save processing time.

```
Name            "Optimized IP XNS ticker bcast filter"
                               # Assign this filter in the multicast path
                               # of a port only--this is very important.
                               #
                               # XNS FILTERING SECTION (Assuming more XNS traffic)
pushField.a     0              #
pushLiteral.a   0xffffffffffff#
ne                             #
accept                         #
pushField.w     12             # Get the type field of the packet and
                               # place it on top of the stack.
pushTop                        # push copy of type
pushLiteral.w   0x0600         # Put the type value for XNS on top of
                               # the stack.
eq                             # If the two values on the top of the
                               # stack are equal, then return a non-zero
                               # value.
pushLiteral.w   0x76c          # Put the lowest socket value on top of
                               # the stack.
pushField.w     30             # Put the value of the socket from the
                               # packet on top of the stack.
ge                             # Compare if the value of the socket is
                               # greater than or equal to lower bound.
pushLiteral.w   0x0898         # Put the highest socket value on top of
                               # the stack.
pushField.w     30             # Put the value of the socket from the
                               # packet on top of the stack.
lt                             # Compare if the value of the socket is
                               # less than the upper bound
and                            # "and" together with "ge" and "lt" test
                               # to determine if the socket value is
                               # "within" the range. If it is, place a
                               # "one" on the stack
                               #
and                            # Compare if XNS & in range
reject                         # reject if XNS and in  range
                               #
                               # IP FILTERING SECTION
                               #
                               # The type field of the packet was
                               # place on top of the stack by the PushTop command.
                               #
pushLiteral.w   0x0800         # Put the type value for IP on top of
                               # the stack.
ne                             # not IP
accept                         # go no further
                               #
pushLiteral.w   0x76c          # Put the lowest socket value on top of
                               # the stack (1900).
pushField.w     24             # Put the value of the socket from the
                               # packet on top of the stack.
ge                             # Compare if the value of the socket is
                               # greater than or equal to lower bound.
pushLiteral.w   0x0898         # Put the highest socket value on top of
                               # the stack (2200).
pushField.w     24             # Put the value of the socket from the
                               # packet on top of the stack.
lt                             # Compare if the value of the socket is
                               # less than the upper bound
and                            # "and" together with "ge" and "lt".
                               # Test to determine if the socket value is
                               # "within" the range. If it is in range,
                               # place a "one" will on the stack.
not                            # Discard (IP & in range)
```

16 IP ROUTING

This chapter provides guidelines and other key information about how to configure a Multilayer Switching Module to route packets using the Internet Protocol (IP). The chapter covers these topics:

- Routing Overview
- Key Concepts
- Routing Models: Port-based and VLAN-based
- Key Guidelines for Implementing IP Routing
- Address Resolution Protocol (ARP)
- ARP Proxy
- Internet Control Message Protocol (ICMP)
- ICMP Redirect
- Broadcast Address
- Directed Broadcast
- Routing Information Protocol (RIP)
- Routing Policies
- Domain Name System (DNS)
- User Datagram Protocol (UDP) Helper
- Standards, Protocols, and Related Reading

For information about how to perform IP multicast routing, see Chapter 18. For information about Open Shortest Path First (OSPF), see Chapter 19.

The management interfaces display "`cb9000`" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.

You can manage IP routing in either of these ways:

- *From the* `ip` *menu of the Administration Console. (See the* Switch 4007 Command Reference Guide.) *You can use the Administration Console after you log in to the system and connect to a slot that houses a Multilayer Switching Module.*
- *From the IP folder of the Web Management software. (See the* Switch 4007 Getting Started Guide.*)*

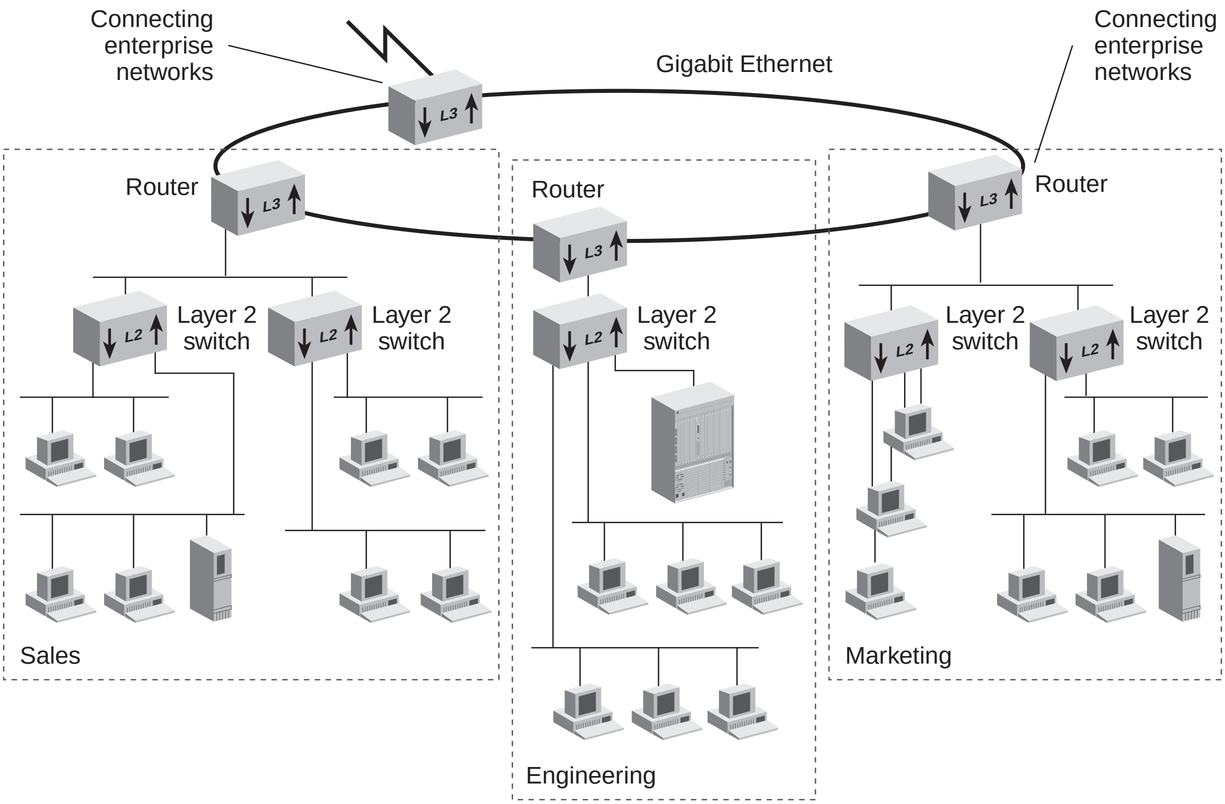
Routing Overview

Routing distributes packets over potentially dissimilar networks. A router is the device that accomplishes this task. Your module, as a Layer 3 device, can act as a router. Routers typically:

- Connect enterprise networks.
- Connect subnetworks (or client/server networks) to the main enterprise network.

Figure 27 shows where routers are typically used in a network. Routing connects subnetworks to the enterprise network, providing connectivity between devices within a workgroup, department, or building.

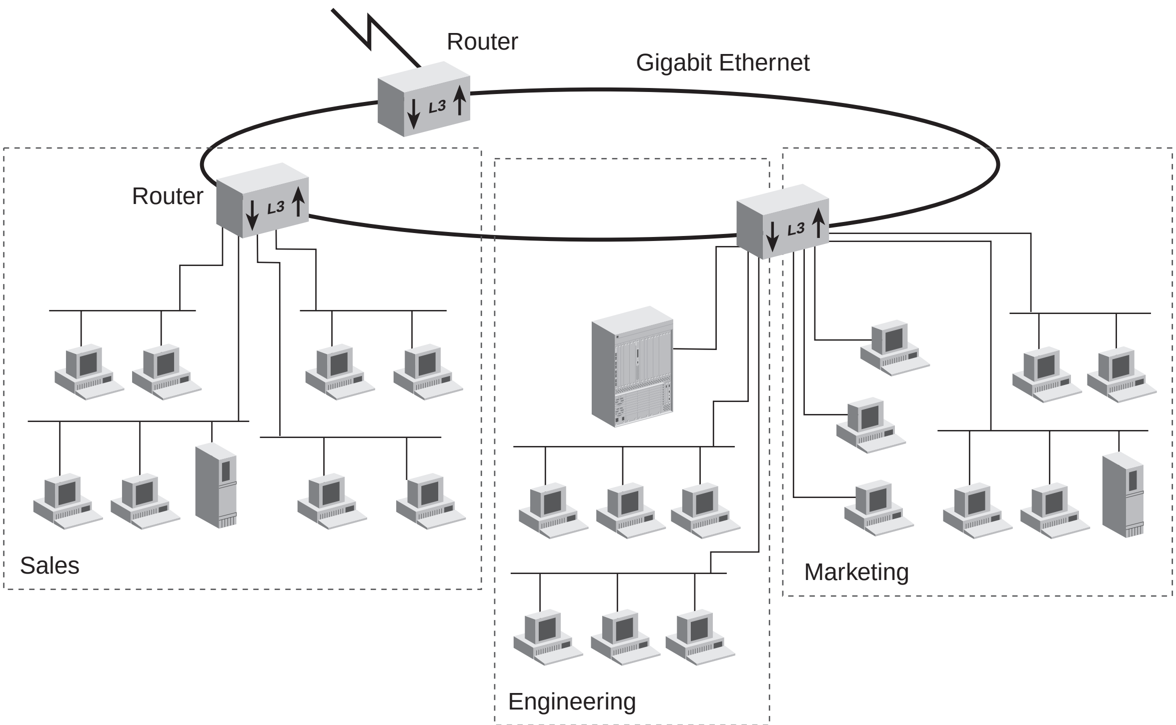
Figure 27 Typical Routing Architecture



Routing in a Subnetworked Environment

Use your system to fit Ethernet switching capability into subnetworked (subnetted) environments. When you put your system into such a network, the system streamlines your network architecture by *routing* traffic between subnetworks and *switching* within subnetworks. See Figure 28.

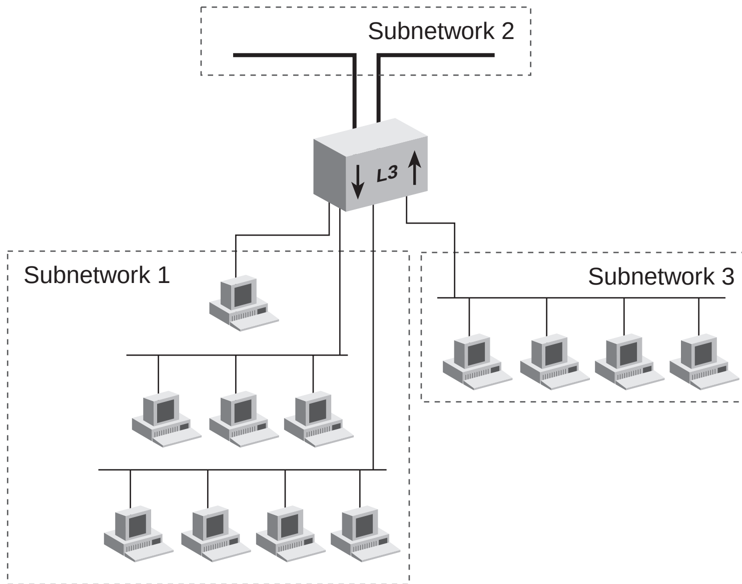
Figure 28 Subnetwork Routing Architecture



Integrating Bridging and Routing

Your module integrates bridging and routing. You can assign multiple ports to each subnetwork. See Figure 29.

Figure 29 Multiple Ethernet Ports Per Subnetwork



Bridging switches traffic between ports that are assigned to the same subnetwork. Traffic traveling to different subnetworks is routed using one of the supported routing protocols. For information about implementing bridging, see Chapter 9.

Bridging and Routing Models

Your module implements routing differently from the way bridges and routers usually coexist.

- Traditionally, network systems first try to route packets that belong to recognized protocols; all other packets are bridged.
- In the 3Com model, the Multilayer Switching Module first tries to determine if the frame is to be switched or routed. If the destination MAC address is not an internal MAC address, then the frame must be routed. If the destination MAC address is an internal MAC address, the frame is further examined to determine if the frame can be switched according to the IEEE 802.1D protocol.

3Com Bridging and Routing

The destination MAC address determines whether the module bridges or routes a packet. Before a host system sends a packet to another host, the host system compares its own network address to the network address of the other host as follows:

- If network addresses are on the same subnetwork, the packet is bridged directly to the destination address of the host.
- If network addresses are on different subnetworks, the packet must be routed from one to the other. In this case, the host sends an ARP request for its default gateway MAC address, then transmits the packet using the MAC address of the default gateway.

Figure 30 illustrates bridging on a 3Com Multilayer Switching Module:

1. The packet enters the module.
2. The bridging layer examines the destination MAC address of the packet. The destination MAC address does *not* correspond to the MAC address of one of the module ports that are configured for routing.
3. The bridging layer selects a segment (port) based on the destination MAC address and forwards the packet to that segment.

Figure 30 3Com Bridging

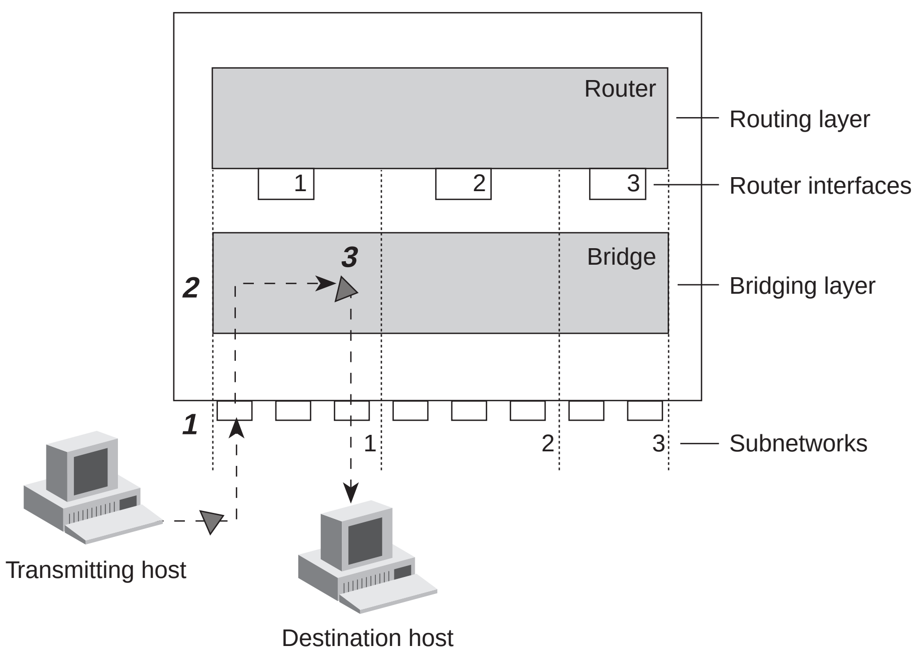


Figure 31 illustrates one 3Com routing model:

1 The packet enters the module.

2 The bridging layer examines the destination address of the packet. The destination address corresponds to the address of one of the module ports that are configured for routing (as opposed to a learned end station address).

3 The packet is passed to the router interface that is associated with the port where the packet was received.

4 The routing layer:

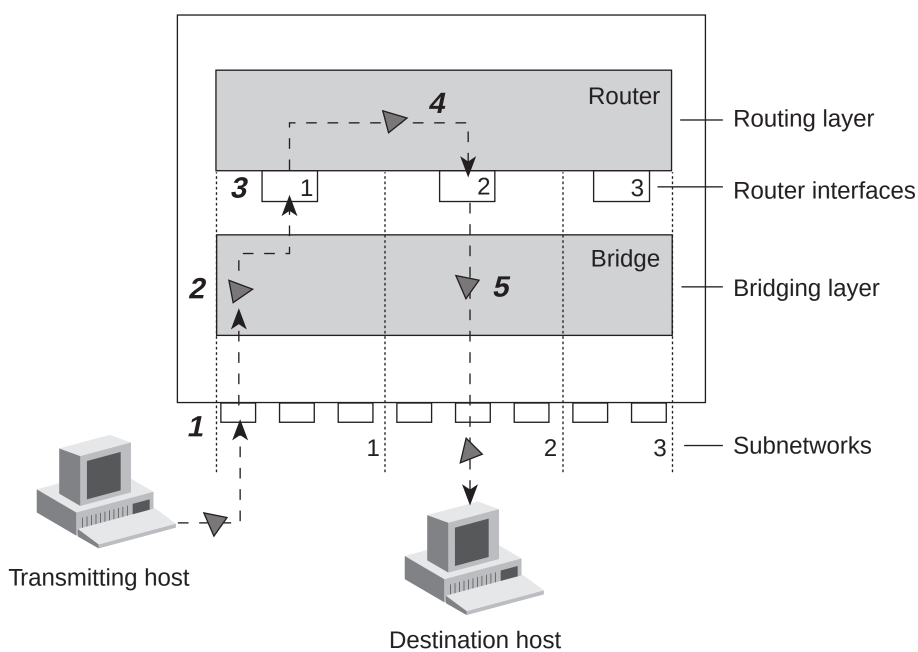
   a Selects a destination interface based on the destination network address

   b Determines the MAC address of the next hop (either the destination host or another gateway)

   c Passes the packet back to the bridging layer

5 The bridging layer then selects a segment (port) based on the destination MAC address and forwards the packet to that segment.

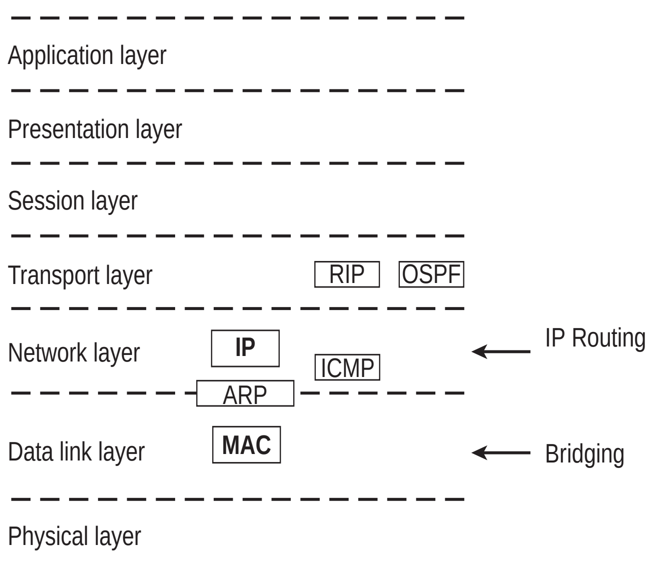
Figure 31 3Com Routing Model



IP Routing Overview

An IP router, unlike a bridge, operates at the network layer of the Open Systems Interconnection (OSI) Reference Model. The network layer is also referred to as Layer 3. An IP router routes packets by examining the network layer address (IP address). Bridges use data link layer MAC addresses to perform forwarding. See Figure 32.

Figure 32 OSI Reference Model and IP Routing



When an IP router sends a packet, it does not know the complete path to a destination — only the next hop (the next device on the path to the destination). Each hop involves three steps:

1. The IP routing algorithm computes the *next hop* IP address and the next router interface, using routing table entries.
2. The Address Resolution Protocol (ARP) translates the next hop IP address into a physical MAC address.
3. The router sends the packet over the network across the next hop.

Features and Benefits

3Com routing in general and IP routing in particular provide the following features and benefits:

- **Economy** — Because you can connect several segments to the same subnetwork with routing, you can increase the level of segmentation in your network without creating new subnetworks or assigning new network addresses. Instead, you can use additional Ethernet ports to expand existing subnetworks. You do not need to create additional subnetworks and assign new network addresses to existing hosts.
- **Optimal routing** — IP routing can be the most powerful tool in a complex network setup for sending devices to find the best route to receiving devices. (The best route here means the shortest and fastest route.)
- **Flexibility** — Using ICMP, you can control the amount, the importance, and the type of traffic on your network.
- **Resiliency** — If a router in the network goes down, the other routers update their routing tables to compensate for this occurrence; in a typical case, there is no need for you to manually intervene.

Key Concepts

IP routers use the following elements to transmit packets:

- Multiple IP Interfaces per VLAN
- Media Access Control (MAC) addresses
- Network-layer addresses
- IP addresses
- Variable Length Subnet Masks (VLSMs)
- Router interfaces
- Routing tables
- Address Protocol Resolution (ARP)
- Internet Control Message Protocol (ICMP)

Multiple IP Interfaces per VLAN

You can overlap IP interfaces without configuring a separate VLAN for each subnet. Multiple IP interfaces can share the same VLAN, allowing multiple subnets to be routed on the same 802.1Q VLAN.

You can define up to 32 IP interfaces on a module. This includes IP routing interfaces for static VLANs, IP VLANs created by router ports or any combination of static VLANs and router port IP VLANs.

If you define multiple interfaces for an IP VLAN, you cannot subsequently modify that IP VLAN to supply Layer 3 address information. If only one routing interface is defined for the IP VLAN, then you can supply Layer 3 address information as long as it matches the Layer 3 information specified for the routing interface. This latter procedure is not recommended, since it makes the IP VLAN a network-based VLAN.

If you continue to use network-based VLANs for this release, you are limited to defining only *one* IP routing interface for that VLAN. When you define an IP routing interface for a static VLAN already configured, the module will not allow you to select a network-based IP VLAN that already has a routing interface defined for it.

Media Access Control (MAC) Address

The MAC address refers to a physical hardware address. On a LAN, the MAC address is the unique hardware number of your device. The MAC address on an Ethernet LAN is the same as your Ethernet address.

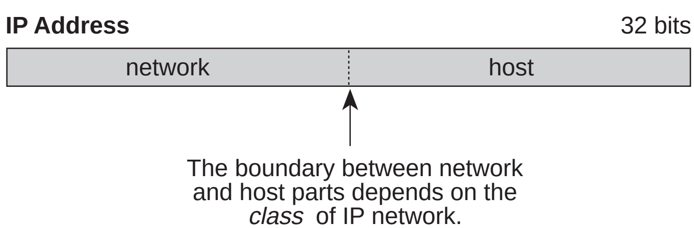
Network-Layer Address

The network-layer address refers to a logical address that applies to a specific protocol. A network-layer address exists at Layer 3 of the OSI reference model.

IP Addresses

IP addresses are 32-bit addresses that consist of a *network part* (the address of the network where the host is located) and a *host part* (the address of the host on that network).

Figure 33 IP Address: Network Part and Host Part



IP addresses differ from Ethernet and Fiber Distributed Data Interface (FDDI) MAC addresses, which are unique hardware-configured 48-bit addresses. A central agency assigns the network part of the IP address, and you assign the host part. All devices that are connected to the same network share the same network part (also called the *prefix*).

Dotted Decimal Notation

The actual IP address is a 32-bit number that is stored in binary format. These 32 bits are segmented into 4 groups of 8 bits — each group is referred to as a *field* or an *octet*. Decimal notation converts the value of each field into a decimal number, and the fields are separated by dots.

Figure 34 Dotted Decimal Notation for IP Addresses

10011110.01100101.00001010.00100000 = Binary notation

158.101.10.32 = Decimal notation

The decimal value of an octet whose bits are all 1s is 255.

Network Portion

The location of the boundary between the network part and the host part depends on the class that the central agency assigns to your network. The three primary classes of IP addresses are A, B, and C:

- **Class A address** — Uses 8 bits for the network part and 24 bits for the host part. Although only a few Class A networks can be created, each can contain a very large number of hosts.
- **Class B address** — Uses 16 bits for the network part and 16 bits for the host part.
- **Class C address** — Uses 24 bits for the network part and 8 bits for the host part. Each Class C network can contain only 254 hosts, but many such networks can be created.

The high-order bits of the network part of the address designate the IP network class. See Table 71.

Table 71 How Address Class Corresponds to the Address Number

Address Class	High-order Bits	Address Number (Decimal)
A	0nnnnnnn	0-127
B	10nnnnnn	128-191
C	11nnnnnn	192-254

Subnetwork Portion

The IP address can also contain a *subnetwork part* at the beginning of the host part of the IP address. Thus, you can divide a single Class A, B, or C network internally, allowing the network to appear as a single network to other external networks. The subnetwork part of the IP address is visible only to hosts and gateways on the subnetwork.

When an IP address contains a subnetwork part, a *subnet mask* identifies the bits that constitute the subnetwork address and the bits that constitute the host address. A subnet mask is a 32-bit number in the IP address format. The *1* bits in the subnet mask indicate the network and subnetwork part of the address. The *0* bits in the subnet mask indicate the host part of the IP address, as shown in Figure 35.

Figure 35 Subnet Masking

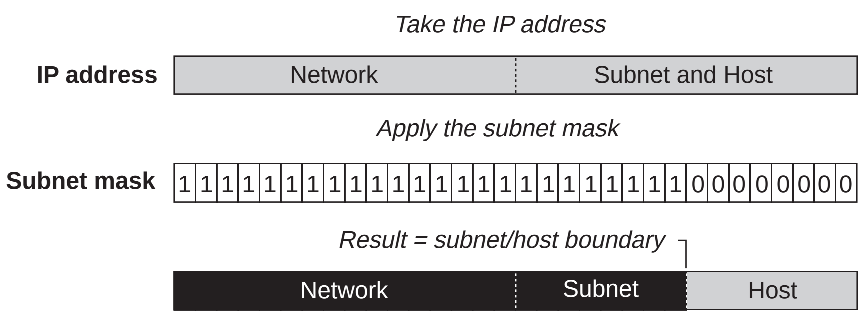


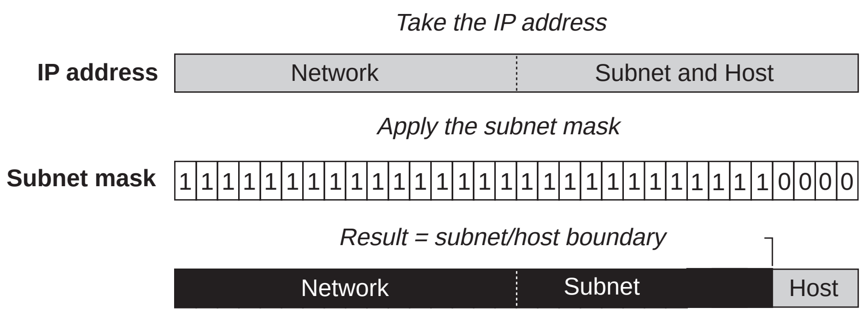
Figure 36 shows an example of an IP address that includes network, subnetwork, and host parts. Suppose the IP address is *158.101.230.52* with a subnet mask of *255.255.255.0*. Since this is a Class B address, this address is divided as follows:

- *158.101* is the network part
- *230* is the subnetwork part
- *52* is the host part

As shown in this example, the 32 bits of an IP address and subnet mask are usually written using an integer shorthand. This notation translates four consecutive 8-bit groups (octets) into four integers that range from 0 through 255. The subnet mask in the example is written as 255.255.255.0.

Traditionally, subnet masks were applied to octets in their entirety. However, one octet in the subnet mask can be further subdivided so that part of the octet indicates an *extension* of the network number, and the rest of the same octet indicates the host number, as shown in Figure 36.

Figure 36 Extending the Network Prefix



Using the Class B IP address from Figure 35 (158.101.230.52), the subnet mask is 255.255.255.240.

The number that includes both the Class B natural network mask (255.255) and the subnet mask (255.240) is sometimes called the *extended network prefix*.

Continuing with the previous example, the subnetwork part of the mask uses 12 bits, and the host part uses the remaining 4 bits. Because the octets are actually binary numbers, the number of subnetworks that are possible with this mask is 4,096 (2^{12}), and the number of hosts that are possible in each subnetwork is 16 (2^4).

Subnet Mask Numbering

An alternate method to represent the subnet mask numbers is based on the number of bits that signify the network portion of the mask. Many Internet Service Providers (ISPs) now use this notation to denote the subnet mask. See Table 72.

Table 72 Subnet Mask Notation

Standard Mask Notation	Network Prefix Notation
100.100.100.100 (255.0.0.0)	100.100.100.100/8
100.100.100.100 (255.255.0.0)	100.100.100.100/16
100.100.100.100 (255.255.255.0)	100.100.100.100/24

The subnet mask 255.255.255.255 is reserved as the default broadcast address.

Variable Length Subnet Masks (VLSMs)

With Variable Length Subnet Masks (VLSMs), each subnetwork under a network can use its own subnet mask. Therefore, with VLSM, you can get more subnetwork space out of your assigned IP address space.

How VLSMs Work

VLSMs get beyond the restriction that a single subnet mask imposes on the network. One subnet mask per IP network address fixes the number of subnetworks and the number of hosts per subnetwork.

For example, if you decide to configure the 158.100.0.0/16 network with a /23 extended-network prefix, you can create 128 subnetworks with each having up to 510 hosts. If some of the subnetworks do not need that many hosts, you would assign many host IP addresses but not use them.

With VLSMs, you can assign another subnet mask, for instance, /27, to the same IP address. So you can assign a longer subnet mask that consequently uses fewer host IP addresses. As a result, routing tables are smaller and more efficient.

This method of further subdividing addresses using VLSMs is being used increasingly more as networks grow in size and number. However, be aware that this method of addressing can greatly increase your network maintenance and the risk of creating erroneous addresses unless you plan the addressing scheme properly.

Guidelines for Using VLSMs

Consider the following guidelines when you implement VLSMs:

- When you design the subnetwork scheme for your network, do not estimate the number of subnetworks and hosts that you need. Work from the top down until you are sure that you have accounted for all the hosts, present and future, that you need.
- Use Open Shortest Path First (OSPF) to carry the extended network prefix information with each route advertisement.

- Make sure that the routers forward routes based on what is known as the *longest match*.

  For example, assume that the destination IP address of a packet is 158.101.26.48 and that the following four routes are in the routing table:

  - 158.101.26.0/24
  - 158.101.3.10/16
  - 158.101.26.32/16
  - 158.95.80.0/8

  The router selects the route to 158.101.26.0/24 because its extended network prefix has the greatest number of bits that correspond to the destination IP address of the packet.

See RFCs 1219 and 1878 for information about understanding and using VLSMs.

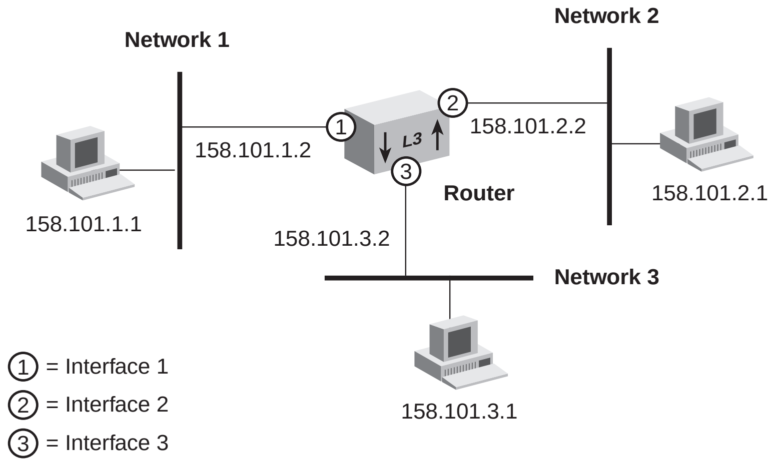
Router Interfaces

A router interface connects the router to a subnetwork. On your Multilayer Switching Module, more than one port can connect to the same subnetwork.

Each router interface has an IP address and a subnet mask. This router interface address defines both the number of the network to which the router interface is attached and its host number on that network. A router interface IP address serves two functions:

- For sending IP packets to or from the router
- For defining the network and subnetwork numbers of the segment that is connected to that interface

Figure 37 Routing Interfaces



To gain access to the module using TCP/IP or to manage the module using the Simple Network Management Protocol (SNMP), set up an IP interface to manage your system and at least one virtual LAN (VLAN). See Chapter 14 for information about how to define a VLAN.

Routing Table

With a routing table, a router or host determines how to send a packet toward its ultimate destination. The routing table contains an entry for every learned and locally defined network. The size of the routing table is dynamic and can hold at least 25,600 entries; the actual number depends upon what other protocols are being routed.

A router or host uses the routing table when the destination IP address of the packet is not on a network or subnetwork to which it is directly connected. The routing table provides the IP address of a router that can forward the packet toward its destination.

The routing table consists of the following elements:

- **Destination IP address** — The destination network, subnetwork, or host.
- **Subnet mask** — The subnet mask for the destination network.
- **Metric** — A measure of the distance to the destination. In the Routing Information Protocol (RIP), the metric is the number of hops through routers.

- **Gateway** — The IP address of the router interface through which the packet travels on its next hop.
- **Status** — Information that the routing protocol has about the route, such as how the route was put into the routing table.
- **Time-to-live (TTL)** — Time-to-live measured in seconds before this learned route will time out.

Figure 38 shows the routing table contents of the router in Figure 37.

Figure 38 Sample Routing Table

Routing table

Destination	Subnet mask	Metric	Gateway	Status	TTL
default route	255.255.255.0	2	160.1.1.254	learned - RIP	170
158.101.1.0	255.255.255.0	2	160.1.1.254	learned - OSPF - INTRA	---
158.101.2.0	255.255.255.0	2	160.1.1.254	learned - OSPF - INTRA	---
158.101.3.0	255.255.255.0	2	160.1.1.254	learned - OSPF - INTRA	---

Routing table data is updated statically or dynamically:

- **Statically** — You manually enter static routes in the routing table. Static routes are useful in environments where no routing protocol is used or where you want to override some of the routes that are generated with a routing protocol. Because static routes do not automatically change in response to network topology changes, manually configure only a small number of reasonably stable routes. Static routes do not time out, but they can be learned.
- **Dynamically** — Routers use a protocol such as RIP or OSPF to automatically exchange routing data and to configure their routing tables dynamically. Routes are recalculated at regular intervals. This process helps you to keep up with network changes and allows the module to reconfigure routes quickly and reliably. Interior Gateway Protocols (IGPs), which operate within networks, provide this automated method.

Default Route

In addition to the routes to specific destinations, a routing table can contain a *default route*. The router uses the default route to forward packets that do not match any other routing table entry.

A default route is often used in place of static routes to numerous destinations that all have the same gateway IP address and interface number. The default route can be configured statically, or it can be learned dynamically.

A drawback to implementing a default static route is that it is a single point of failure on the network. You can implement Virtual Router Redundancy Protocol (VRRP) on your network to remedy this problem. For more information about VRRP, see Chapter 17.

Routing Models: Port-based and VLAN-based

There are two basic routing models for implementing how a bridge and a router interact within the same 3Com switch. They are:

- Port-based routing (routing versus bridging)

  The module first tries to route packets that belong to recognized protocols, and all other packets are bridged. When you configure a port-based IP interface, the port ignores the spanning tree state even if the port state is set to blocking.

- VLAN-based routing (routing over bridging)

  The module first tries to determine if the frame will be switched or routed. The module does this by examining the destination MAC address:

  - If the destination MAC address is not an internal MAC address, then the frame must be switched and is forwarded according to the IEEE 802.1D protocol.
  - If the destination MAC address is an internal MAC address, the frame is further examined to determine if the frame is a routed frame (Layer 3) or a request to the switch itself (Layer 2).

  This model allows the module to give the frame first to Layer 2 to be bridged by the VLAN, and then given to the router only if the frame cannot be bridged. This scheme gives you the flexibility to define router interfaces on top of several bridge ports.

Your module, as a routing device, has the ability to implement either type of routing scheme, "routing over bridging" and "routing versus bridging". Each kind of routing scheme requires its own interface type:

- **Routing over Bridging requires a VLAN-based IP Interface** — A VLAN-based interface requires you to first configure a VLAN and then create a router interface over that VLAN.
- **Routing versus Bridging requires a Port-based IP Interface** — A port-based interface requires you to configure a router interface on top of a single physical port.

Key Guidelines for Implementing IP Routing

To route network traffic using IP, you must perform these tasks in the following order:

1. Configure Trunks (Optional)
2. Configure IP VLANs
3. Establish Your IP Interfaces
4. Enable IP Routing

Configure Trunks (Optional)

Trunks (also known as aggregated links) work at Layer 2 and allow you to combine multiple Fast Ethernet or Gigabit Ethernet into a single high-speed link between two switches.

If you intend to use trunking on an IP device, configure your trunks *before* you set up VLANs and IP interfaces. In this case, you must specify the anchor port (the lowest-numbered port) to associate with the trunk. For example, if ports 7 through 12 are associated with a trunk, specifying 7 to 12 defines the VLAN to include all of the physical ports in the trunk (ports 7 through 12).

For more information about trunking, see Chapter 12.

Configure IP VLANs

If you want to use IP routing, you must first configure the VLAN to use IP. An IP VLAN is called a *protocol-based VLAN*.

Protocol-based VLANs such as IP VLANs group one or more switch ports together for one or more specified Layer 3 protocols. You can also create network-based VLANs, which are IP VLANs that are grouped according to the IP network address and mask.

When you use allClosed VLAN mode on a Multilayer Switching Module in your system, you can enable the module to ignore Spanning Tree Protocol (STP) mode on a per-VLAN basis; that is, ignore STP blocked ports. (When STP detects multiple paths to a destination, it blocks all but one of the paths.) You can use Ignore STP mode to avoid disruptions to routing connectivity, based on the STP state.

See Chapter 14 in this guide to learn about VLANs.

Establish Your IP Interfaces

To establish an IP interface:

1. Determine your interface parameters.
2. Define the IP interfaces.

Interface Parameters

Each IP routing interface has these standard characteristics:

- **IP address** — An address from the range of addresses that the Internet Engineering Task Force (IETF) assigns to your organization. This address is specific to your network and Multilayer Switching Module.
- **Subnet mask** — The 32-bit number that uses the same format and representation as an IP address. The subnet mask determines which bits in the IP address are interpreted as the network number/subnetwork number and the host number. Each IP address bit that corresponds to a 1 in the subnet mask is in the network/subnetwork part of the address. Each IP address bit that corresponds to a 0 is in the host part of the IP address.
- **State** — The status of the IP interface. It indicates whether the interface is available for communications (`up`) or unavailable (`down`).
- **VLAN interface index (for in-band management)** — The number of the IP VLAN that is associated with the IP interface. When the

Layer 3 module prompts you for this option, the menu identifies the available VLAN indexes.

Important Consideration

Consider the following issue before you establish an IP interface:

- Before you assign IP addresses, map out the entire network and subnetwork IP addressing scheme. Plan for future expansion of address numbers as well.

The `ip interface define` (in-band) and `management ip interface define` (out-of-band) options are documented in the *Command Reference Guide*. To learn how to use the Web Management Console to set up IP interfaces, see the *Switch 4007 Getting Started Guide*.

Defining an IP Interface

After you determine the VLAN index, IP address, and subnet mask for each IP interface, you can define each interface. Use the Administration Console or the Web Management Console to define an IP interface.

Remember that you must define a VLAN and select IP as a protocol that the VLAN supports before you define the IP (routing) interface. VLANs are described in Chapter 14*.*

To define your IP interface, decide if and how to implement the following IP features:

- ARP proxy
- ICMP Redirect
- ICMP Router Discovery
- Broadcast address
- Directed broadcast
- RIP
- Routing policies
- DNS
- UDP Helper

These features are discussed later in this chapter.

Enable IP Routing

To enable IP routing, use the `ip routing` command on the Administration Console or use the IP Configuration form in the Web Management software. By default, IP routing is disabled on the Multilayer Switching Module.

You can use the Routing Information Protocol (RIP) or the Open Shortest Path First (OSPF) protocol to take advantage of routing capabilities. RIP is discussed in this chapter; OSPF is discussed in Chapter 19*.*

Administering IP Routing

Keep these points in mind while you administer the IP network:

- Flush the ARP cache regularly if you set the age time to 0.
- Set up a default route.

  The Multilayer Switching Module uses the default route to forward packets that do not match any other routing table entry. You may want to use the default route in place of routes to numerous destinations that all have the same gateway IP address. If you do not use a default route, ICMP is more likely to return an `address not found` error.
- Before you can define static routes, you must define at least one IP interface. See "Defining an IP Interface" earlier in this chapter for more information. Remember the following guidelines:
  - Static routes remain in the routing table until you remove them or the corresponding interface.
  - Static routes take precedence over dynamically learned routes to the same destination.
  - Static routes are included in periodic RIP updates sent by your Layer 3 module.

Address Resolution Protocol (ARP)

ARP is a low-level protocol that locates the MAC address that corresponds to a given IP address. This protocol allows a host or router to use IP addresses to make routing decisions while it uses MAC addresses to forward packets from one hop to the next.

You do not need to implement ARP — the module has ARP capability built in, but you can manipulate and display the contents of the ARP cache.

When the host or router knows the IP address of the *next* hop towards the packet destination, the host or router translates that IP address into a MAC address before sending the packet. To perform this translation, the host or router first searches its *ARP cache*, which is a table of IP addresses with their corresponding MAC addresses. Each device that participates in IP routing maintains an ARP cache. See Figure 39.

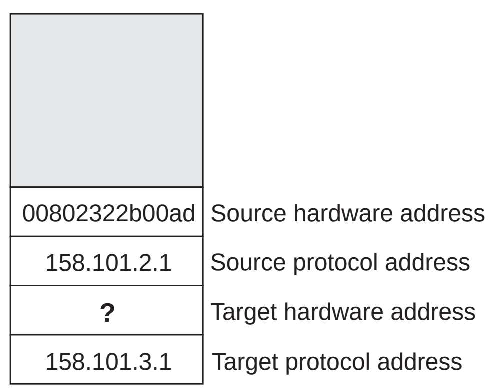
Figure 39 Example of an ARP Cache

ARP cache

IP address	**MAC address**
158.101.1.1	00308e3d0042
158.101.2.1	0080232b00ab

If the IP address does not have a corresponding MAC address, the host or router broadcasts an *ARP request* packet to all the devices on the network. The ARP request contains information about the target and source addresses for the protocol (IP addresses). See Figure 40.

Figure 40 Example of an ARP Request Packet



When devices on the network receive this packet, they examine it. If their address is not the target protocol address, they discard the packet. When a device receives the packet and confirms that its IP address matches the

target protocol address, the receiving device places its MAC address in the target hardware address field and sends the packet back to the source hardware address.

When the originating host or router receives this *ARP reply*, it places the new MAC address in its ARP cache next to the corresponding IP address. See Figure 41.

Figure 41 Example of ARP Cache Updated with ARP Reply

ARP cache

IP address	MAC address
158.101.1.1	00308e3d0042
158.101.2.1	0080232b00ab
158.101.3.1	0134650f3000

After the MAC address is known, the host or router can send the packet directly to the next hop.

Important Considerations

Keep the following things in mind about this protocol:

- Enter a static ARP entry when the ARP resolution does not result in an ARP entry in the cache. For example, some applications do not respond to ARP requests and, consequently, specific network operations may time out for lack of address resolution.
- Enter a static ARP entry in a test environment if your test analyzer cannot respond to an ARP request.
- Setting an ARP cache age time of zero (no aging) is useful in the middle of lengthy tests so that ARP requests do not have to be issued.

  If you do set an ARP cache age time of zero, be aware that the ARP cache can quickly grow in size and consume module resources. In this case, be sure to flush the ARP cache after your tests are complete.
- You can keep ARP cache entries if you refresh the ARP cache; otherwise, the Multilayer Switching Module removes the entries after they reach their defined age time.

ARP Proxy

ARP proxy allows a host that has no routing ability to determine the MAC address of a host on another network or subnet.

When ARP proxy is enabled and a workstation sends an ARP request for a remote network, the module determines if it has the best route and then answers the ARP request by sending its own MAC address to the workstation. The workstation then sends the frames for the remote destination to the module, which uses its own routing table to reach the destination on the other network.

Important Considerations

Consider the following issues with ARP proxy:

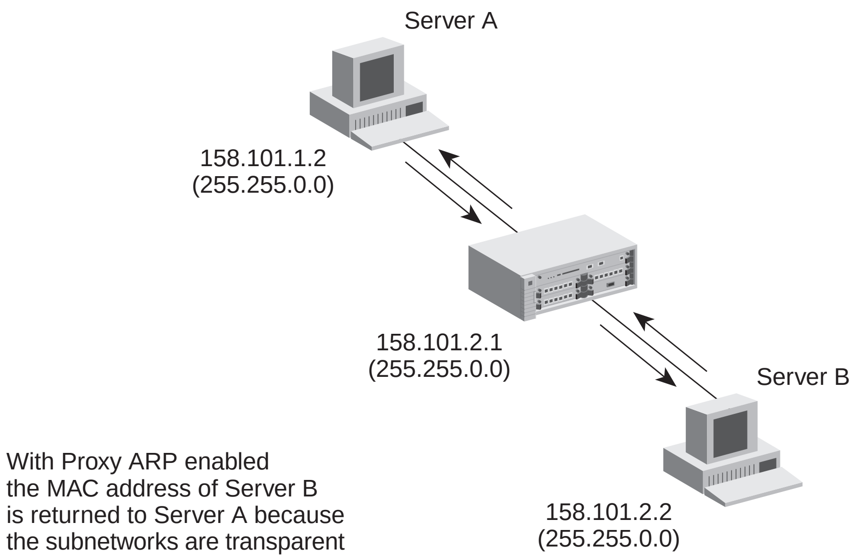
- Do not use ARP proxy if you are using VLSMs because ARP proxy works by seeing the entire network configuration as one network.
- ARP proxy increases ARP traffic to handle the increased mapping of IP addresses to MAC addresses.

Example

In the following example, Server A cannot use the router as a gateway to Server B (if ARP proxy is disabled) because Server A has its subnet mask set to broadcast (using ARP) its IP network address as 158.101.0.0, while the IP network address of the router is 158.101.1.0.

However, if the router has ARP proxy enabled, the router answers the request of Server A with its own MAC address — thus, all traffic sent to Server B from Server A is addressed to the corresponding IP interface on the router and forwarded appropriately.

Figure 42 ARP Proxy



Internet Control Message Protocol (ICMP)

Because a router knows only about the next network hop, it is not aware of problems that may be closer to the destination. Destinations may be unreachable if:

- Hardware is temporarily out of service.
- You specified a nonexistent destination address.
- The routers do not have a route to the destination network.

To help routers and hosts discover problems in packet transmission, a mechanism called Internet Control Message Protocol (ICMP) reports errors back to the source when routing problems occur. With ICMP, you can determine whether a delivery failure resulted from a local or a remote problem.

ICMP performs these tasks:

- **Creates more efficient routing (*ICMP Redirect*)** — Often the host route configuration specifies the minimum possible routing data that is needed to communicate (for example, the address of a single router). The host relies on routers to update its routing table. In the process of routing packets, a router may detect that a host is not using the best route. The router sends an ICMP Redirect to this host, requesting that the host use a different gateway when it sends packets to a destination. The host then sends packets to that destination using the new route if it is able to interpret ICMP Redirect directives.
- **Uses the router with the highest preference level as the default gateway (*ICMP Router Discovery*)** — ICMP Router Discovery is useful if you have multiple gateways that connect a particular subnet to outside networks. By using the preference setting, you can select which gateway is the preferred choice.

For more information about ICMP Redirect and ICMP Router Discovery, see "Internet Control Message Protocol (ICMP)" and "ICMP Router Discovery" later in this chapter.

ICMP Router Discovery

ICMP Router Discovery directs a host to use the router with the highest preference level as the default gateway. ICMP does this by enabling hosts that are attached to multicast or broadcast networks to discover the IP addresses of their neighboring routers and determine which router to use for a default gateway. If you prefer, you can make this default gateway choice yourself.

Important Considerations

Keep the following points in mind with ICMP Router Discovery:

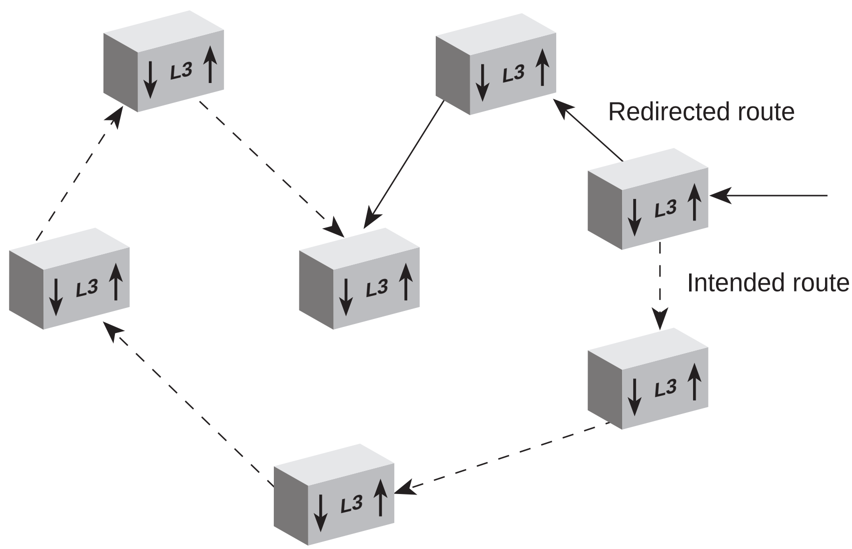
- You need not manually configure a default route.

  Although IP traffic may initially be directed to any of the routers on the LAN, ICMP Redirect messages subsequently channel IP traffic to the correct router.
- ICMP Router Discovery is useful on large networks, or when the network topology has undergone a recent change.
- If you are on a small network that is relatively stable, consider using a static route to the gateway instead of ICMP Router Discovery to reduce network traffic.

- The minimum value hex 80000000 directs neighboring hosts not to use the address, even though it may be advertised as a default router address. It may be useful to configure an address with a preference level of hex 80000000 (rather than setting its Advertise flag to `false`) when you use advertisements for “black hole” detection. (See RFC 1256 for detailed information about this term.)

Example Figure 43 shows how ICMP can dynamically determine a router to act as the default gateway.

Figure 43 ICMP Router Discovery



See the documentation for your workstation to determine whether you can configure your workstation to use this protocol.

See RFC 1256 for detailed information about ICMP Router Discovery.

ICMP Redirect

ICMP Redirect adds another layer of intelligence to routing. ICMP Redirect:

- Informs the sending device of the frame that there is a more efficient route to the destination.
- Routes the frame via the more efficient route.

Use the Administration Console or the Web Management software to enable ICMP Redirect.

Important Considerations

Keep the following things in mind with ICMP Redirect:

- ICMP Redirect determines if the sending interface is the same as the receiving interface.
- ICMP Redirect determines if the source device of the frame is on a direct-connect network.
- You can enable or disable ICMP Redirect on a per-interface basis.
- There is a performance cost associated with this redirect activity. You have to monitor the activity to gauge its effect on the network.
- Performance can be affected if the sending device ignores the recommendations of ICMP Redirect, in which case the performance cost of ICMP Redirect is incurred while the benefits are wasted.
- If you disable ICMP Redirect, the hardware routes the frame, and no messages are sent back to the sending device. At some point, however, the number of retries associated with less intelligent hardware routing overtake any benefits that are associated with the speedier routing that hardware provides.
- To maximize the effectiveness of ICMP Redirect, have ICMP Redirect on the module that is connected to the greatest number of other routing devices.
- Disable ICMP Redirect if you have overlapped IP interfaces on ports that are not configured to use 802.1Q VLAN tagging. Doing so provides better routing performance between the overlapped subnets.
- If you have two interfaces that belong to different VLANs that share a given port and you want to completely disable ICMP redirects for that port, disable the redirects for *each* interface that shares that port. If you disable ICMP Redirect for only one interface and enable it for the other, you may not get the performance improvement that you want.

Broadcast Address

You can set a broadcast address for each defined IP interface. Your module uses this broadcast address when forwarding directed broadcast packets, and when advertising RIP packets.

When you define an IP interface, the broadcast address is 255.255.255.255. This is the default address.

Important Considerations

Keep the following points in mind when you use broadcast address:

- You cannot change the broadcast address for an IP interface if you have already defined any RIP advertisement addresses.
- If you are concerned with security, filter all inbound and outbound broadcast traffic. Many hosts are set up to respond to an echo request to their broadcast address with an echo reply, which can breach security.

Directed Broadcast

A directed broadcast contains 1s in the host portion of the address field. You can choose to have your module, on a per-interface basis, enable or disable the forwarding of directed broadcast frames.

Important Considerations

Keep the following points in mind when you use directed broadcast:

- When your module receives a directed broadcast and the destination is different from the interface on which it was received:
  - Your module forwards the directed broadcast if directed broadcast is `enabled`
  - Your module drops the directed broadcast if directed broadcast is `disabled`
- Set the directed broadcast to reflect your security requirements. If you have a critical IP interface, disabling directed broadcast can, for example, protect against denial-of-service attacks by malicious users.

Routing Information Protocol (RIP)

RIP is the protocol that implements routing. RIP does this by using Distance Vector Algorithms (DVAs) to calculate the route with the fewest number of hops to the destination of a route request. Each device keeps its own set of routes in its routing table. RIP is an Interior Gateway Protocol (IGP) for TCP/IP networks.

RIP operates using both active and passive devices.

- *Active* devices, usually routers, broadcast RIP messages to all devices in a network or subnetwork and update their internal routing tables when they receive a RIP message.
- *Passive* devices, usually hosts, listen for RIP messages and update their internal routing tables, but do not send RIP messages.

An active router sends a broadcast RIP message every 30 seconds. This message contains the IP address and a metric (distance) from the router to each destination in the routing table. In RIP, each router through which a packet must travel to reach a destination counts as one network *hop*.

Basic RIP Parameters

RIP has several parameters to consider when you set up RIP to use in your network. When you configure an IP interface, the module already has the RIP parameters set to the defaults listed in Table 73.

Table 73 RIP Parameters

RIP Parameter	Default Value
RIP Mode	learn
Cost	1
Poison Reverse	enabled
Advertisement Address	limited broadcast address (255.255.255.255)

RIP Mode

The four available settings for RIP mode are as follows:

- **Disabled** — The Multilayer Switching Module ignores all incoming RIP packets and does not generate any RIP packets of its own.
- **Learn** — The Multilayer Switching Module processes all incoming RIP packets, but it does not transmit RIP updates.
- **Advertise** — The Multilayer Switching Module broadcasts RIP updates, but it does not process incoming RIP packets.
- **Enabled** — The Multilayer Switching Module broadcasts RIP updates and processes incoming RIP packets.

Compatibility Mode

The RIP-1 compatibility mode determines how the software sends periodic RIP-2 updates. (RIP-1 always uses the advertisement list when sending RIP-1 advertisements.)

- When the module is configured to advertise RIP-2 packets and compatibility mode is `disabled`, the software uses the multicast address of 224.0.0.9 when sending periodic updates. Doing so reduces the load on hosts that are not configured to listen to RIP-2 messages.
- When the module is configured to advertise RIP-2 packets and compatibility mode is `enabled`, the software uses the advertisement list for RIP-2 updates.

Cost

You can use RIP to calculate the route metrics (the *cost)* for you. The cost is the number of hops that the packet needs to get to its destination. The RIP cost is a number between 1 and 15. (A number higher than 15 is not allowed, because RIP cannot negotiate more than 15 hops.)

Most facilities assign a cost of 1 to all interfaces. However, if you have two links with differing speeds, such as a dial-up link versus a direct link, you may want to raise the cost of the dial-up link so that the direct link is more likely to be used.

Poison Reverse

Poison Reverse is a RIP feature that you use specifically with a scheme called *Split Horizon*. The module enables Poison Reverse by default.

Split Horizon avoids the problems that reverse-route updates can cause. Reverse-route updates are sent to a neighboring router and include the routes that are learned from that router. Split Horizon omits the routes that are learned from one neighbor in the updates that are sent to that neighbor (the reverse routes).

Poison Reverse is essentially another layer of protection against advertising reverse routes.

- When you enable (default mode) Poison Reverse, the Multilayer Switching Module advertises reverse routes in updates, but it sets the metrics to 16 (infinity). Setting the metric to infinity breaks the loop immediately when two routers have routes that point to each other.
- When you disable Poison Reverse, such reverse routes are not advertised.

You can disable Poison Reverse because it augments what Split Horizon already does, and it puts additional information that you may not need into RIP updates.

Advertisement Address

The module uses the advertisement address specified to advertise routes to other stations on the same network. The module uses this address for sending updates.

Each interface that you define initially uses the default broadcast address (255.255.255.255) as the advertisement address. If you change the broadcast address, the address that you specify becomes the new RIP advertisement address.

Effects and Consequences

After you add an advertisement address, you cannot subsequently change the broadcast address.

RIP-1 Versus RIP-2

Like RIP-1, RIP-2 allows the module to dynamically configure its own routing table. RIP-2 is much more flexible and efficient than RIP-1, however, because RIP-2 advertises using the multicast method, which can advertise to a subset of the network (RIP-1 uses the broadcast method, which advertises to the whole network). RIP-2 can do this because it includes a subnet mask in its header.

If your module receives a RIP-2 packet, your module puts the route into the routing table with the subnet mask that was advertised.

Important Considerations

Consider the following issues when you implement RIP on your module:

- Use RIP-2 rather than RIP-1 if possible, because RIP-2 uses subnet masking and the next hop field. Subnet mask advertising allows you to use VLSM. (See "Variable Length Subnet Masks (VLSMs)" earlier in this chapter for more information.)
- Set RIP as follows:
  - **RIP-1** — `learn`
  - **RIP-2** — `enabled`

  In this way, the module keeps track of the RIP-1 and RIP-2 address routes in its routing table and forwards the routes as well.
- 3Com recommends that you not advertise RIP-1 and RIP-2 together. If you do, two different sets of IP addresses may go into to the routing table for every one RIP advertisement, which quickly reduces the efficiency of the routing table.

Routing Policies

IP routing policies allow you to control how routes are sent from and received by the routing table in your module. Both RIP and OSPF have routing policy capabilities. This section describes the RIP routing policies; OSPF routing policies are discussed in Chapter 19.

There are two basic types of routing policies:

- **Import policies** — Import policies control what routes are added to the routing table. (That is, the import policies control which routes your module can accept from other routers.) When RIP or OSPF forwards a route to the routing table, the router searches its import policies before adding the route to the routing table.
- **Export policies** — Export policies control what routes from the routing table are advertised by the RIP and OSPF protocols to other routers. (That is, export policies control which routes your module can forward to other routers.) When RIP or OSPF are preparing a route advertisement, the router searches its export policies before advertising the route to the network.

You can create up to 128 routing policies total. The total is shared between OSPF and RIP policies.

Routing policies can control the entire flow of routing information among the network, the protocols, and the routing table manager.

Routing Policies are often referred to as Route Filters because defining policies for accepting and forwarding routes is very much like defining filters to screen which routes may be forwarded or accepted.

How Routing Policies Work

Each router keeps a table of current routing information, called the *routing table*. The router protocols on the module receive routes from or advertise routes to the network.

When a route needs to be added to the routing table:

1. The protocol (OSPF or RIP) that receives the route sends that route to the routing table manager.
2. The routing table manager searches the Import policies.
3. If the import policy allows the route to be accepted, the routing table manager adds the route to the routing table; otherwise, the route is discarded.

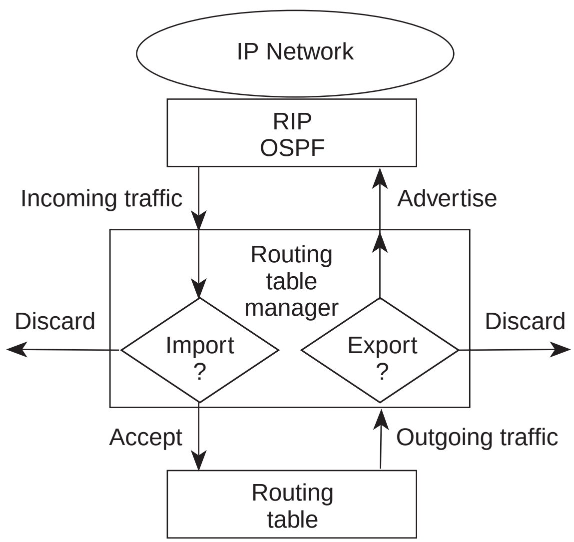
The router also needs to periodically advertise routes to other routers:

1. The protocol (OSPF or RIP) polls the routing table manager for routes to advertise to other routers.
2. The routing table manager searches the Export policies.
3. If the export policy allows the route to be advertised, the routing table manager advertises the route on the network; otherwise, the route is not sent.

Figure 62 shows the first level of decision-making in routing policies. Routing policies also contain two parameters that help further refine this system: metrics and administrative weight.

- **Metric (cost) adjustment** — Specifies how many hops to assign to the route. The range of the metric is 0 through 16 hops. (If you specify 0, the module does not modify the metric; if you specify 16, you are specifying that the route is unreachable — 16 represents infinity.)
- **Administrative weight** — Controls the relative weight of each policy with respect to another policy. The range extends from 1 to 16, with 16 taking the greatest precedence.

Figure 44 IP Routing Policies



Important Considerations

Even though Routing Policies are not true routing protocols and are considered optional, they can increase network efficiency.

- You can increase speed *and* security simply by limiting the number of devices from which the router receives data.
- You can establish a neighbor list of devices, which is a list of trusted systems whose addresses you have confidence in.
- You can associate the list of devices with a specific traffic direction, either incoming or outgoing. As a result, you can assign very precise routes of traffic, thereby keeping tight control over them.
- By adjusting the relative importance of certain policies over others, you can exercise great control over the type and amount of traffic to and from your system.

Implementing RIP Routing Policies

RIP routing policies determine which RIP routes can be accepted into the routing table, and which RIP and OSPF routes can be advertised.

RIP Metric Adjustments

You can use the following arithmetic operators to adjust the RIP metrics:

Table 74 RIP Metric Adjustments

Metric	Description
+nn	Increase metric by nn
-nn	Decrease metric by nn
*nn	Multiple metric by nn
/nn	Divide metric by nn
%nn	Modulus — returns the remainder of the metric

CAUTION: *Use caution if you use arithmetic operators to adjust the relative value of the number of hops that you allow a route to have; you can inadvertently make a route unreachable.*

RIP Import Policy Conditions for Specified Interfaces

Table 75 lists the policy conditions for RIP import policies:

Table 75 RIP Import Policy Conditions

Source Router	Route (address/mask)	Action	Description
Specified router	Specified route/mask	accept	Accept specified route from specified source router on specified interfaces with or without metric adjustments (+, -, *, /, %).
Specified router	all (0.0.0.0)	accept	Accept all routes from specified router on specified interfaces with or without metric adjustments (+, -, *, /, %).
all (all routers)	Specified route/mask	accept	Accept specified route on specified interfaces with or without metric adjustments (+, -, *, /, %).
all	all	accept	Accept all routes on specified interfaces with or without metric adjustments (+, -, *, /, %).
Specified router	Specified route/mask	reject	Reject specified route from specified router on specified interfaces. (Metrics do not apply because the route itself is rejected.)
Specified router	all	reject	Reject all routes from specified router on specified interfaces.
all	Specified route/mask	reject	Reject specified route from all routers on specified interfaces.
all	all	reject	Reject all routes on specified interfaces.

RIP Export Policy Conditions for Specified Interfaces

Table 76 lists the policy conditions for the RIP export policies:

Table 76 RIP Export Policy Conditions

Protocol	Source Router	Route	Action	Description
RIP, OSPF, static	Specified router or all routers	Specified route/mask	accept	Advertise RIP/OSPF/static specified route from specified source router on specified interfaces with or without metric adjustments (+, -, *, /, %).
RIP, OSPF, static	Specified router or all routers	all (0.0.0.0)	accept	Advertise all RIP/OSPF/static routes from specified router on specified interfaces with or without metric adjustments (+, -, *, /, %).
RIP, OSPF, static	Specified router or all routers	Specified route/mask	reject	Do not advertise the RIP/OSPF/static specified route on specified interfaces.
RIP, OSPF, static	Specified routers or all routers	all	reject	Do not advertise all RIP/OSPF/static routes on specified interfaces.

Multiple Matched Routing Policies

Because you can use a wildcard parameter (`all`) to specify a source or target route, there are times when several policies can apply to the same route.

When the module perceives that there is more than one policy for the same route, it follows this hierarchy of rules to resolve the policy conflict:

- The policy with the highest administrative weight
- The policy that matches the specific source
- The policy that matches the most number of bits for the route
- The policy that matches the origin protocol
- The policy with the lowest index

Setting Up RIP Routing Policies

To configure a routing policy, follow these general steps:

1. Establish an Export policy that controls the advertisement of routes through RIP, regardless of the source from which the route is learned.
2. Establish an Import policy that accepts or refuses to accept information on routes learned by RIP from a trusted neighbor.
3. To control the reporting of routes that are learned from specific sources, establish the following policies:
   - Export policy for routes learned from OSPF
   - Static policy for reporting static (user-configured) routes

   If you decide to have routes reported with a metric that is calculated from the routing table, you can manipulate the conversion formula that RIP uses to convert a routing table metric into one that RIP understands.

4. Establish a policy to report OSPF routes so that the metrics that are reported with these routes are imported into RIP without being changed.

Effects and Consequences

Consider the following points when you use routing policies:

- Configure the administrative weight setting carefully because this setting has the highest priority in resolving policy conflicts.
- If you use routing policies, do not implement static routes. Routing policies work with routes that are updated dynamically.
- Use routing policies only if you need the security, or if you need more control over the routing tables than other IP features, such as VLSMs, give you.
- To control whether a route is accepted or forwarded without making specific changes to your network configuration, consider setting the Cost metric as high as possible, and the administrative weight as low as possible.

Creating RIP Routing Policies

To set a routing policy, you need to know the following parameters:

- **Policy type** — The determination whether to accept a route into the routing table (import) or advertise a route from the routing table (export)
- **Source address** — The routing device that is sending the route to your module
- **Route address** — The actual device IP address of the route origin
- **Route subnet mask** — The subnet mask of the device IP address of the route origin
- **IP interface** — The IP interface on your module that the route is coming in on
- **Policy action** — The determination whether to accept or reject the route
- **Metric adjustment** — The determination to increase or decrease the route metric (the number of hops) for the route
- **Administrative weight** — The level of importance of this policy: 1 is low priority, 16 is high priority

The policy takes effect on the selected interfaces only if the origin protocol matches the protocol that is enabled for the selected interfaces.

Sketch out a topology of your routers and the proposed routing policies of each to get an understanding of how the routers work together and how traffic flows.

Table 77 lists the import policies for Router B:

Table 77 Router B Routing Policies

Policy Type	Source Address	Route Address	Route Subnet Mask	IP Interface	Policy Action	Metric	Weight
Import	10.1.2.2	130.1.0.0	255.255.0.0	1	accept	1	1
Import	10.1.2.2	131.1.0.0	255.255.0.0	1	reject	–	2
Import	10.1.2.2	132.1.0.0	255.255.0.0	1	reject	–	1
Import	10.1.2.2	133.1.0.0	255.255.0.0	1	accept	1	2

In this example, only routes 130.1.0.0 and 133.1.0.0 are accepted into the routing table of Router B.

Domain Name System (DNS)

The Domain Name System (DNS) client allows you to specify a hostname rather than an IP address when you perform various operations (for example, when you use `ping` or `traceRoute` to contact an IP station).

With DNS, you can specify one or more name servers that are associated with a domain name. Each name server maintains a list of IP addresses and their associated host names. When you use `ping` or `traceRoute` with a hostname, the DNS client attempts to locate the name on the name servers that you specify. When the DNS client locates the name, it resolves it to the associated IP address.

You can resolve an IP address to a host name or a host name to an IP address on a name server. Enter either the host name or the IP address; the DNS client displays the pair.

Important Considerations

When you set up DNS servers on your LAN, remember the following:

- Always set up more than one DNS name server (a primary and secondary server) so that the lookup service does not have a single point of failure.
- If your ISP changes the Classes of Internetwork Service, change the DNS settings on each host that the ISP services.

See UNIX Network File System (NFS) documentation for information about how to create and maintain lists of domain names and IP addresses on the name servers.

See Chapter 17 *and also the* Command Reference Guide*, for information about how to use* `ping` *and* `traceRoute`*.*

User Datagram Protocol (UDP) Helper

User Datagram Protocol (UDP) Helper allows TCP/IP applications to forward broadcast packets from your module (as a router) and to another part of the network.

Two common uses of the UDP Helper feature are:

- **Bootstrap Protocol (BOOTP)**

  Using BOOTP through a logical port, you can boot a host through the router, even if the host is on another part of the network. UDP packets that rely on the BOOTP relay agent are modified and then forwarded through the router.

- **Dynamic Host Configuration Protocol (DHCP)**

  Using DHCP, a host can retrieve its own configuration information, including the IP address, from a DHCP server through the IP network. DHCP makes it easier to administer the IP network. With DHCP, you can dynamically configure a host with new information.

3Com implements a generic UDP Helper agent in the module that can apply to any UDP port.

Implementing UDP Helper

You have to set the following UDP Helper parameters:

- **UDP port number** — A logical address, not a port (interface) on your module. BOOTP (including DHCP) uses UDP port 67.
- **IP forwarding address** — The IP address to which the packets are forwarded. You can have up to 63 combinations of port numbers and IP forwarding addresses per router. You can also have multiple IP address entries for the same ports.
- **Hop count** — The number of interfaces that the module uses to forward a packet through the router.
- **Threshold** — The maximum number of times that the module forwards a packet to the network. By default, there is no BOOTP relay threshold. (The default value is `0`.)

The commands to implement these parameters are described in the "IP Routing" chapter of the *Command Reference Guide*.

You need to have a thorough understanding of your network configuration to use UDP Helper. Review the network topology before you implement UDP Helper.

Configuring Overlapped Interfaces

Overlapped IP interfaces are multiple logical interfaces that are defined for a single physical port. You can specify how UDP Helper forwards packets from overlapped IP interfaces with one of these interface options:

- **First** — The module uses the first overlapped IP interface of the port as the source network for forwarded packets.
- **Even** — The module hashes the MAC address of the client to determine the source network for forwarded packets. This arrangement evenly distributes the interface among those on the network.
- **Sequential** — The module assigns each overlapped IP interface, in turn, as the source network for forwarded packets.

You can view the UDP Helper configuration when you configure the forwarding address.

Important Considerations

Consider the following points when you use UDP Helper:

- The maximum BOOTP hop count (how many steps the module uses to forward a packet through the router) is 16; the default hop count limit is 4. Keep the hop count as low as possible for performance purposes.
- 3Com recommends that you keep the UDP port number at 67. The port number 67, which is the industry standard, helps ensure that UDP packets do not get dropped due to an unknown destination failure.
- You can always add or remove a port number or IP forwarding address defined for UDP Helper.

Standards, Protocols, and Related Reading

This section describes how to obtain more technical information about IP.

Requests For Comments (RFCs)

Documents called Requests for Comments (RFCs) contain information about the entire set of protocols that make up IP. Some of the RFCs that pertain to the discussions in this chapter are:

- **RFC 791** — Internet Protocol
- **RFC 1219** — Subnetwork Numbers
- **RFC 951, 1542** — UDP Helper
- **RFC 1878** — VLSMs
- **RFC 1256** — ICMP Router Discovery Messages
- **RFC 1058** — RIP
- **RFC 1723** — RIP Version 2
- **RFC 1786** — IP Routing Policies
- **RFC 2400** — Internet Official Protocol Standards

You can obtain RFCs from the Internet using the following URL:

`http://sunsite.auc.dk/RFC`

Standards Organizations

Standards organizations ensure interoperability, create reports, and recommend solutions for communications technology. The most important standards groups are:

- International Telecommunications Union (ITU)
- Electronic Industry Association (EIA)
- American National Standards Institute (ANSI)
- International Standards Organization (ISO)
- Institute of Electrical and Electronic Engineers (IEEE)
- Internet Engineering Task Force (IETF)
- National Institute of Standards and Technology (NIST)

Related Reading

For more information about the IP protocol suite, see the following books:

- *High Speed Networks: TCP/IP and ATM Design Principles.* William Stallings, Prentice Hall, 1998
- *Local Area Networks: Architectures and Implementations*. James Martin, Prentice Hall, 1994
- *Internetworking with TCP/IP: Principles, Protocols, and Architecture*. Douglas Comer, Prentice Hall, 1995

17 VIRTUAL ROUTER REDUNDANCY PROTOCOL (VRRP)

The Virtual Routing Redundancy Protocol (VRRP) can prevent a loss of network operations for end hosts due to the failure of the static default IP gateway. VRRP accomplishes this by allowing you to designate a number of other routers as Backup routers in the event that the Master router (the default router) should fail for any reason.

Topics covered in this chapter include:

- VRRP Overview
- Key Concepts
- Important Considerations
- VRRP and Other Networking Operations
- Using VRRP On Your Switch 4007
- Configuring VRRP
- Standards, Protocols, and Related Reading

Before you implement VRRP, be sure that you have a good understanding of how IP networks function. See Chapter 16 for more information about IP networks. Also, be sure to read this chapter thoroughly before you set up VRRP on your network.

After you log in to the system and connect to a slot that houses a Multilayer Switching Module, you can manage VRRP in these ways:

- *From the* `vrrp` *menu of the Administration Console. (See the* Switch 4007 Command Reference Guide.*)*
- *From the VRRP folder of the Web Management software. (See the* Switch 4007 Getting Started Guide.*)*

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

VRRP Overview

A critical component of IP networking is the way in which hosts and routing devices find the next-hop address in a connectionless environment. There are several different ways of determining the next-hop address, but they all fall into two basic categories:

- Router to Router
- Host to Host and Host to Gateway

Router to Router

Router-to-router communication is usually accomplished by means of a routing protocol such as Routing Information Protocol (RIP) or Open Shortest Path First (OSPF), or by static routes. Routers consult their own routing tables to make intelligent next hop decisions for the forwarding of IP packets.

Host to Host and Host to Gateway

IP host-to-host communication typically begins with an ARP request to the destination host address, providing that the destination resides on the same subnet as the sending device. If the destination address resides on a non-local subnet, then the sending device must use one of the following methods to learn the route to the remote network:

- Routing Protocols
- ICMP Router Discovery
- Static Route
- Default Gateway

Routing Protocols

Routing protocols provide dynamic updates to end stations in the event of a network failure, but they are typically not used on most hosts because they require additional setup, processing power and, in some cases, additional software.

ICMP Router Discovery

ICMP Router Discovery directs a host to use the router with the highest preference level as the default gateway. Internet Control Message Protocol (ICMP) does this by enabling hosts that are attached to multicast or broadcast networks to discover the IP addresses of their neighboring routers and determine which router to use for a default gateway. If you prefer, you can make this default gateway choice yourself.

Static Route

A static route is an IP address that is user-configured and fixed. Static routes are useful if the host only needs to access a few networks; in this case, static routes actually require less overhead than dynamic routing protocols. However, in today's networking environment in which traffic patterns are less predictable, many routes are usually required, and static routes then become prohibitive to maintain.

Default Gateway

Most host stations today use a default gateway to facilitate routing. You simply define for the host an IP address on the local subnet of a router that is responsible for routing packets to their destinations. This approach is widely deployed today; however, it has one major drawback: if the default gateway becomes unavailable, then all routing to remote networks stops, requiring manual intervention to restore connectivity even if there are alternate paths available.

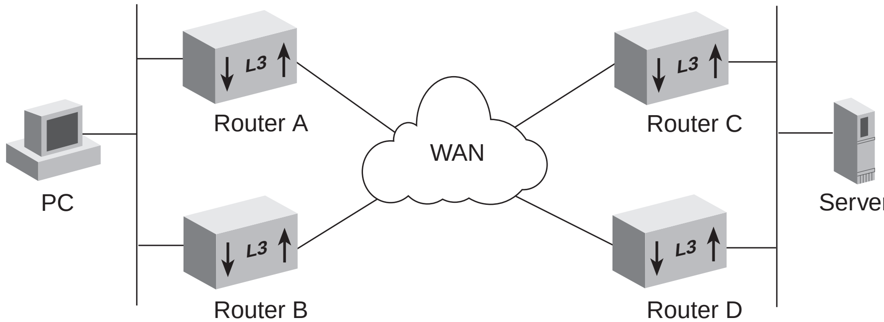
VRRP addresses this drawback by defining an election protocol that dynamically assigns responsibility for a *virtual router* to one of the VRRP routers on a LAN. The election process automatically detects a failure of the primary (Master) router, and transfers all traffic forwarding to the backup router. All of this is done without your intervention, which dramatically increases uptime in an IP network.

Example

In the simplest scenario, a VRRP configuration includes two routers, a primary router (called the Master router) and a backup router. If the Master router fails for any reason, the backup router assumes all forwarding functions for the Master router. The backup router monitors the network for hello packets, which are periodically sent by the Master router (the default time period is 1 second). If the backup router misses three hello packets in succession, that router assumes forwarding functions for the Master.

See Figure 45 for a visual representation of a simple virtual router configuration.

Figure 45 Simple VRRP Configuration



In the example shown in Figure 45, Router A is the default gateway for the workstation labeled PC, which provides access to the Wide Area Network (WAN) and to the device labeled Server. Assume that no router discovery protocols have been configured and that the default gateway is static.

If the workstation loses its connection to Router A, the workstation loses all remote connectivity because its default gateway is no longer available. However, if VRRP is enabled in this same scenario, Router B detects the loss of connectivity to Router A, and Router B assumes all forwarding responsibilities on behalf of Router A. This transfer of forwarding responsibilities allows the workstation to have continued access across the WAN to the server.

Key Concepts

This section contains some VRRP definitions that you should know before reading further.

- **VRRP router** — A router running the VRRP protocol. A VRRP router can:
  - Act as a Master router with actual addresses on a interface.
  - Act simultaneously as a Backup for other routers with additional virtual router mappings and priorities for those routers.
- **Virtual router** — A logical entity, managed by VRRP, that acts as the default router for hosts on a shared LAN. The virtual router has a unique identifier called the Virtual Router Identifier (VRID), and has a set of associated IP addresses across the LAN.

- **Virtual router master** — The VRRP router that forwards packets sent to the IP addresses associated with the virtual router. Also called the Master router. A virtual router is the Master when:
  - You configure it (using the Administration console, the Web Management console, or SNMP) as the primary IP address for a given interface.
  - Backing up a Master that has been disconnected or disabled.
- **Virtual router backup** — The VRRP router that assumes packet forwarding responsibility for a virtual router if the Master fails. Also called the Backup router.
- **IP address owner** — The router that is the original owner of the IP addresses that virtual router incorporates into its own IP address set. The IP address owner must be a VRRP participant.
- **Primary IP address** — An IP address selected from the set of available interface addresses. This is the IP address that VRRP uses in its advertisements that supply the source of the IP packet.
- **Virtual router initialize** — A state in which a virtual router is defined but not enabled. A virtual router is also in the initialize state when its associated interface is not operational.

How VRRP Works

When you assign Master router responsibilities to one of the virtual routers on the LAN, the Master controls the IP addresses associated with a virtual router. The Master router forwards the IP packets sent to the IP addresses it controls.

The backup process works as follows. The Master router sends out periodic VRRP advertisement messages, at time intervals you set, to the other VRRP routers and to the hosts on the common LAN. (A VRRP advertisement consists of the IP addresses that the Master owns and the virtual MAC address of the virtual router.) If the Master stops forwarding advertisements to the other routers for a predetermined period of time, the other routers automatically enter an election process to determine which router takes over Master responsibilities.

After the original Master again become operational, it begins again to broadcast advertisements to the other virtual routers if pre-empt mode is enabled. Packet forwarding responsibility then shifts back to the original Master router.

For this scheme to work, the association between VRIDs and IP addresses must be coordinated among all VRRP routers across the LAN: otherwise, the backup router does not have a valid set of IP addresses to use.

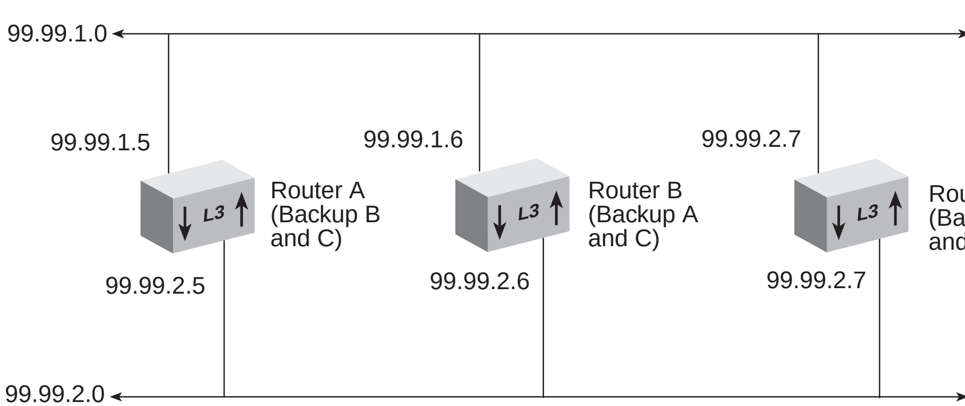
Virtual Router Decision-making

The example in Figure 45 shows only two routers, so there is no ambiguity as far as which router should have assumed responsibility upon a failure. However, there can be more than one virtual router on a network because there can be more than one backup router for each static gateway. This is because a single backup router, at the time of assuming primary router responsibilities, becomes the single point of failure.

See Figure 46 for an example of a network topology that:

- Allows all routers on the LAN to be backed up by more than one virtual router
- Allows hosts on any subnetwork to reach destinations on any other subnetwork in the extended network

Figure 46 Multiple Virtual Routers Backing Up Each Other



The parallel design in Figure 46 takes advantage of the capabilities of VRRP. This design can be extended to include more routers and more subnetworks. In a more complex virtual router scheme with many backup routers, this method ensure that all routers have adequate backup in the event of a failure.

VRRP provides for this by making you assign each virtual router on the LAN a priority value between 1 and 255. (255 means that the virtual router is the actual owner of the IP addresses.) If the Master fails, the virtual router with the next-highest priority takes over Master responsibilities until the original Master comes back online.

If two routers have the same priority, VRRP resolves the conflict by selecting the virtual router with the numerically-highest primary IP address. In other words, if Virtual Router A (primary IP address of 1.1.1.2) and Virtual Router C (primary IP address of 1.1.1.3) both have a priority of 100, Virtual Router C would have a higher priority than Virtual Router A.

CAUTION: *Configure all of the routers participating in the VRRP scheme on your network to have the same representation of the network. If some routers have a different view of the topology than others, a backup router failure is more likely, with the resultant loss of some or all end hosts' connection to the network.*

Important Considerations

This section provides information to be aware of when you implement VRRP:

- The Master router forwards the IP addresses that you have associated with the primary virtual router, and:
  - Responds to ARP requests for the IP address or addresses that are associated with the virtual router.
  - Forwards packets that have a destination Link Layer MAC address that matches the virtual router MAC address. In other words, the Master forwards packets that hosts have sent to the virtual router to be routed.
  - Discards packets addressed to the IP address or addresses associated with the virtual router if the virtual router is not the IP address owner. Otherwise, ping, SNMP, and Telnet do not function properly.
  - Sends periodic VRRP advertisement messages. (Set the advertising interval to be short enough to provide a timely transition to another router should the Master fail. Try an advertising interval of 1 second.)

- A Backup router monitors the availability and state of the Master, and:
  - Does not respond to ARP requests for the IP address or addresses associated with the virtual router.
  - Discards packets that have a destination Link Layer MAC address that matches the virtual router MAC address.
  - Does not accept packets addressed to the IP address or addresses associated with the virtual router
- Hosts obtain the virtual router's MAC address by means of an ARP broadcast.
- VRRP is *not* a routing protocol, and its usefulness is only as good as the design of the network upon which it is implemented. Good network design is critical in ensuring the success of router redundancy.
- The virtual routers must be on the same VLAN.
- VRRP supports Proxy ARP; the virtual router uses the virtual router MAC address in Proxy ARP replies.
- VRRP supports Fiber Distributed Data Interface (FDDI) and Ethernet
- Consider using VRRP in conjunction with port-based routing to provide router redundancy on your campus backbone. See Chapter 16 for an example of port-based routing on a campus backbone.

VRRP and Other Networking Operations

Read this section for information about how VRRP interacts with other networking functions, including:

- Spanning Tree Protocol (STP)
- Dynamic routing protocols:
  - Routing Information Protocol (RIP)
  - Routing Information Protocol version 2 (RIP-2)
  - Open Shortest Path First (OSPF)
- IP Multicast
- ICMP Redirect
- Quality of Service (QoS)

Spanning Tree Protocol (STP)

Figure 46 earlier in this chapter shows how you can set up VRRP parallel routers to provide total redundancy in your inter-LAN operations. However, because VRRP uses MAC addresses in its advertisements, this topology can represent a bridge loop to STP. In this parallel topology, VRRP advertisements must go out on the network in AllClosed mode with IgnoreSTP (Ignore Spanning Tree Mode) *enabled*.

Carefully evaluate your bridging and routing topologies before you incorporate VRRP into your network operations.

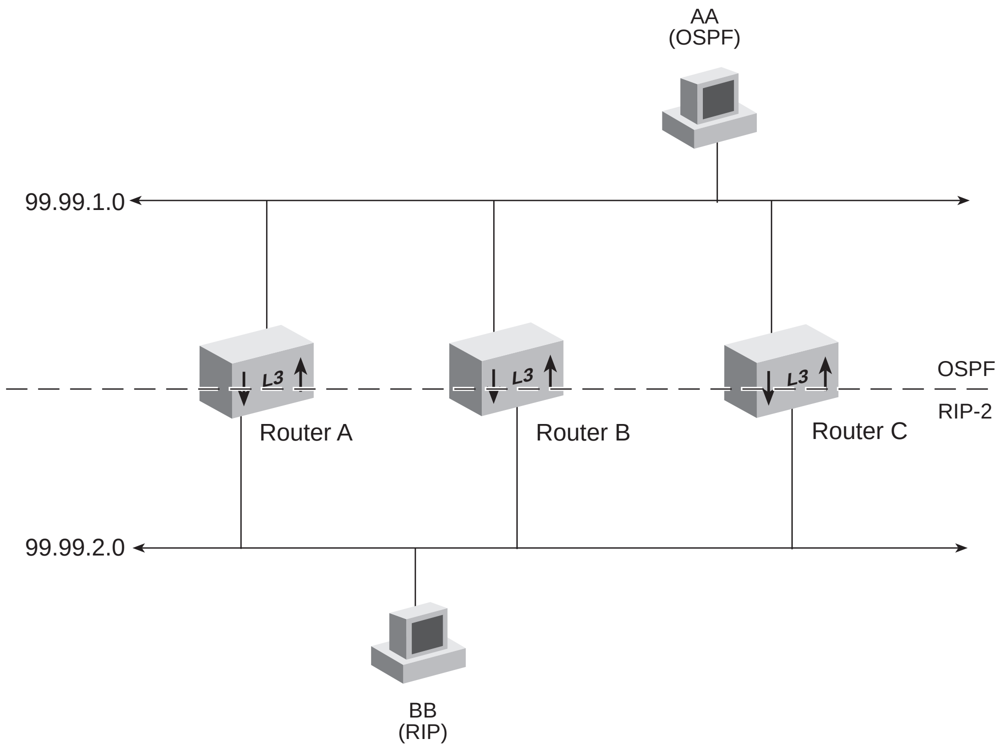
Dynamic Routing Protocols (RIP, RIP-2, OSPF)

The dynamic routing protocols RIP, RIP-2, and OSPF have their own facilities to track routes across networks. You can continue to use these protocols with VRRP routers, but on any given subnetwork, you must configure the same routing protocols with the same parameters.

Figure 47 shows how, in a parallel routing environment, OSPF is configured on each interface in the 99.99.1.0 subnetwork, and RIP-2 is configured on each interface in the 99.99.2.0 subnetwork. The device AA has a gateway of Router A.

If Router A becomes unavailable, Router B can take over because the 99.99.1.0 subnetwork has OSPF configured for each routing interface. If, however, dynamic routing protocols are configured on a router-per-router basis, so that Router B had RIP-2 configured on the router's interface to the 99.99.1.0 subnetwork, the gateway becomes unavailable because of a dynamic routing protocol mismatch.

Figure 47 Proper Use of Dynamic Routing Protocols with VRRP



IGMP Queries

IP multicast routers use IGMP to query subnetworks in order to detect host members of multicast groups. IGMP specifies a querier election process in which one router per subnetwork is designated to issue the IGMP Query messages to host members. The designated router, called the *Querier*, always has the lowest IP address in the subnetwork.

If the Querier goes down, another router can be designated to take its place. The fewer routers that you have designated as possible Queriers, the more efficient the handover is.

Be aware that, if you introduce a parallel router topology to take advantage of VRRP, you can introduce a topology that is not optimal for IGMP operations, especially as the number of routers increase. Carefully gauge the effect of VRRP on your IGMP operations.

ICMP Redirect

Using ICMP Redirect in conjunction with VRRP might cause gateway access problems due to potential conflicts between actual MAC addresses that ICMP Redirect uses and the virtual MAC addresses that VRRP uses. Disable ICMP Redirect if you are using VRRP.

Quality of Service

You can enable Quality of Service (QoS) to run on modules running the VRRP protocol. As with the case of dynamic routing protocols, however, you must configure QoS parameters consistently across the routing interfaces on the same subnetwork.

Also, periodically examine how QoS uses the route cache because the cache may fill up faster in a VRRP environment. Consider classifying specific port ranges in this case.

IP Routing Policies

If you are using IP routing policies to control traffic on your network you must apply the same rate limits to all virtual routers on the LAN, Master as well as Backups. Failure to match routing policies among all virtual routers on the LAN could, for example, leave some routing destinations unreachable.

Dynamic Host Configuration Protocol (DHCP)

Consider using VRRP if your network uses the Dynamic Host Configuration Protocol (DHCP). DHCP provides for a default gateway and an end-host IP address, and therefore is at risk to be a single point of failure.

Using VRRP On Your Switch 4007

This section provides information about configuring VRRP specifically for your Switch 4007. Topics include:

- VRRP with Multiple Virtual Routers
- VRRP Activity
- VRRP with a Single Virtual Router

VRRP with Multiple Virtual Routers

Locally attached end stations (on Layer 2 modules) in the Switch 4007 environment are connected to a router (on a Multilayer Switching Module backplane port) by means of an *internal LAN* (the default VLAN) across the switch fabric module.

See Figure 48 for a graphic example of a VRRP topology on two Switch 4007s. This sample topology directly connects end stations to a backbone Switch 4007 16-slot chassis. The end stations can communicate with end stations on LANs that are external to the 16-slot chassis (in this example, the external LAN is on a Switch 4007 7-slot chassis). End stations exist on several departmental IP subnetworks, with the backbone Multilayer Switching Modules acting as the default gateways for these end stations.

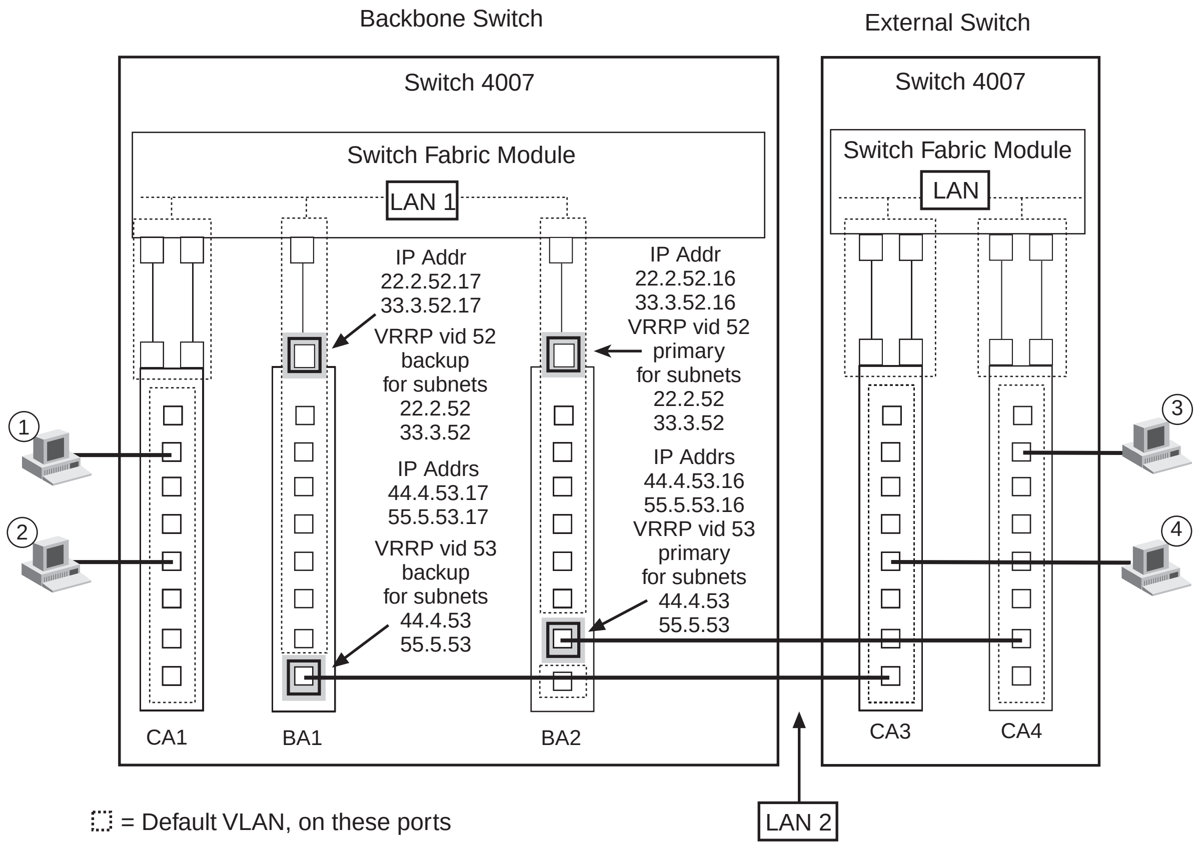
The Multilayer Switching Modules route traffic between locally attached and externally attached hosts using their respective internal and external LANs.

To provide router redundancy for traffic flow between locally and externally attached end stations, this topology requires *two* virtual routers to be configured:

- **LAN1 —** the Primary router on the backplane port of BA2, with the Backup router on the backplane port of BA1.
- **LAN2 —** the Primary router on the front panel port of BA2, with the Backup router on the front panel port of BA1.

To maintain traffic between locally and externally attached end stations in the event of a router failure of only one *of the Virtual Routers from Primary to Backup, RIP-2 or OSPF must be enabled.*

Figure 48 Sample VRRP Topology with More than One Virtual Router



= Default VLAN, on these ports

= Protocol VLAN, Protocol = IP, configured on this port

= VRRP configured on this port

All IP Addresses are submitted to a Class C with a mask of 255.255.255.0

1 IP=22.2.52.1, Gateway=22.2.52.16

2 IP=33.3.52.2, Gateway=33.3.52.16

3 IP=44.4.53.3, Gateway=44.4.53.16

4 IP=55.5.53.4, Gateway=5.5.53.16

Spanning Tree Considerations

To prevent the links from going into blocking mode on the External Switch side, you must:

- Remove the BA1 and BA2 front panel ports in the Protocol VLAN from the Default VLAN.
- Set the Spanning Tree Protocol (STP) state for these ports equal to Remove.

There is no Spanning Tree loop for the following reasons:

- The Layer 3 front panel port drops any non-IP frames received from the external switch.
- The Layer 3 backplane port forwards only routing traffic to the BA1 and BA2 front panel port.

End Station Configuration

The backbone Layer 2 modules are populated with departmental end stations on different subnetworks:

- The backplane port of BA2 is configured with a single Protocol (IP) VLAN, with multiple IP interfaces.
- The backplane port of the BA2 is the default gateway for each departmental subnetwork end station.
- The backplane port of the BA2 functions as a one-armed router for traffic between departmental end stations on Layer 2 modules.

The External Switch is populated with departmental end stations on different subnetworks.

- The front panel port of BA2 is configured with a single Protocol (IP) VLAN, with multiple IP interfaces.
- The front panel port of the BA2 is the default gateway for each departmental subnetwork end station.
- The front panel port of the BA2 functions as a one-armed-router for traffic between departmental end stations on the external switch.

Traffic between end stations on the Backbone Switch 4007 Layer 2 module(s) and end stations on the External Switch is routed between the BA2 backplane and front panel router ports.

VRRP Activity

If the entire BA2 module goes down, then both Backup virtual routers on BA1 switch over to Master and the end stations' ability to address one another on LAN1 (the switch fabric module) and to address the end stations on LAN2 (external switch) is maintained.

However, if only one of the Primary routers goes down, this addressing ability between the LAN1 end stations and the LAN2 end stations is lost *without* RIP-2 or OSPF running.

Sequence of Failover Events

Suppose the link between the backbone BA2 front panel port and the External switch goes down. The following events occur:

1. VRRP switches over to Backup, preserving addressability between end stations on LAN2.
2. The Master router for LAN2 (External Switch) becomes the front panel port of the BA1.
3. The Master router for LAN1 (the switch fabric module) remains the backplane port of the BA2.

However, successful recovery of operations depends upon the status of the dynamic routing protocols (RIP-2 or OSPF):

Without RIP-2 or OSPF: Addressability between the LAN1 end stations and the LAN2 end stations is lost.

- The backplane port of BA2 remains the default gateway for the LAN1-connected end stations.
- The front panel port of BA1 becomes the default gateway for the LAN2-connected end stations.

Problem The backplane port router of BA2 *does not* have the ability to address the 44.4.53 and 55.5.53 subnetworks on the front panel port router of BA1.

With RIP-2 or OSPF: The routing protocol advertises a path to subnetworks 44.4.53 & 55.5.53 through the 22.2.52.17 address, (the backplane port router of the BA1 module). When the front panel router port on BA2 goes down:

- The backplane port router of BA2 updates its routing table from the advertisement of BA1 because the BA2 backplane port router no longer has subnetworks 44.4.53 & 55.5.53 as locally attached networks.
- The backplane port router of BA2 *acquires* addressability to subnetworks 44.4.53 & 55.5.53 by means of the backplane port router of BA1.

The routing table for the backplane port router of BA2 shows 22.2.52.17 as the next hop for the subnetworks 44.4.53 & 55.5.53.

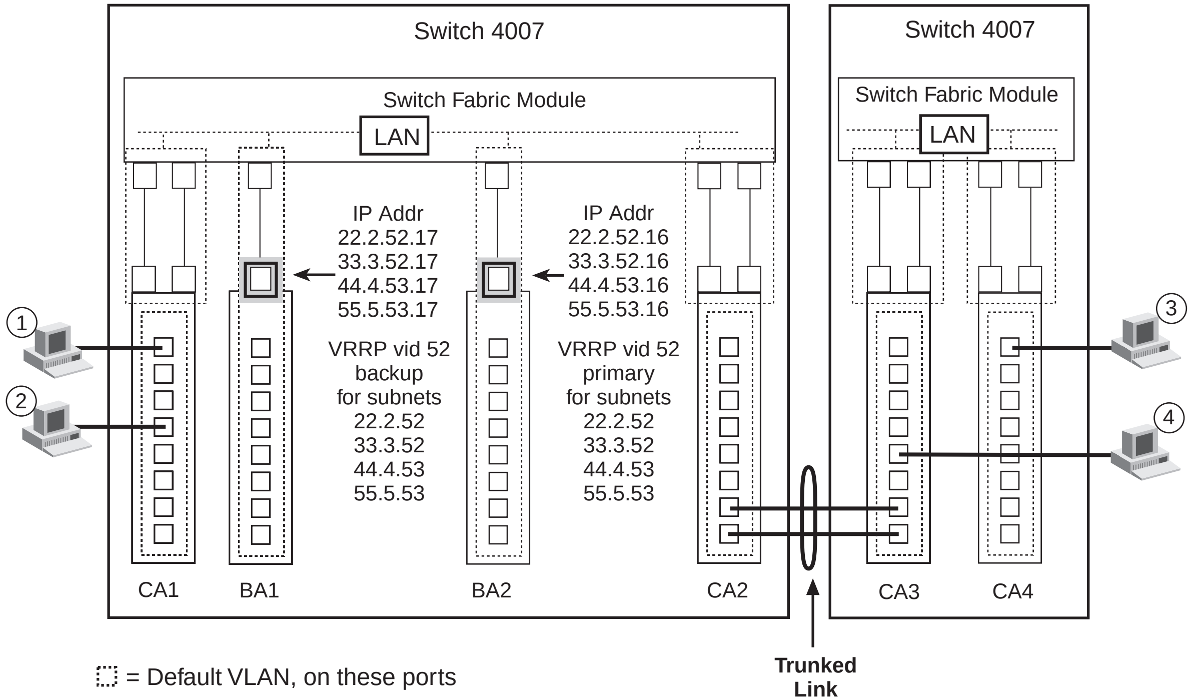
The backplane port of BA1 is still the Backup VRRP router and the backplane port of BA2 is still the Master VRRP router for locally-attached end stations (VRRP VID 52). Traffic flow between end station 1 and end station 2 progresses in the following manner:

Traffic from	**Goes to**
22.2.52.1 (end station 1)	22.2.52.16 (BA2/backplane port default gateway for end station 1)
22.2.52.16 (BA2/backplane port)	22.2.52.17 (BA1/backplane port, next hop to 55.5.53.4)
22.2.52.17 (BA1/backplane port)	55.5.53.17 (BA1/front panel port)
55.5.53.17 (BA1/front panel port)	55.5.53.5 (end station 2)

VRRP with a Single Virtual Router

The topology shown in this section is a variation of the VRRP Topology seen in Figure 48. This topology has a single one-armed router handling routing between subnetworks for both interchassis and intrachassis traffic in a single, flat network.

Figure 49 Sample VRRP Topology with a Single Virtual Router



= Default VLAN, on these ports

= Protocol VLAN, Protocol = IP, configured on this port

= VRRP configured on this port

All IP Addresses are submitted to a Class C with a mask of 255.255.255.0

1 IP=22.2.52.1, Gateway=22.2.52.16

2 IP=33.3.52.2, Gateway=33.3.52.16

3 IP=44.4.53.3, Gateway=44.4.53.16

4 IP=55.5.53.4, Gateway=5.5.53.16

In this topology, there is a single point of failure (because there is a one-armed router) between the left and right chassis. Loss of either the CA2 module in the left chassis, or the CA3 module in the right chassis results in loss of connectivity between the left chassis and the right chassis.

Configuring VRRP

This section provides details about configuring multiple VRRP routers, following the topology in Figure 48. Router 1 is on the Backplane Port of a 12-Port 10/100BASE-TX Fast Ethernet Multilayer Switching Module in Slot 3, and Router 2 is on the Backplane Port of a 10-Port 100BASE-FX Fast Ethernet Layer 3 Switching Module in Slot 5. This involves two general tasks:

- Configuring Router 1 as the Master Router
- Configuring Router 2 as the Backup Router

Configuring Router 1 as the Master Router

Configuring Router 1 as the Master router, as shown in Figure 48, involves the following tasks:

- Configuring the Protocol (IP) VLAN of the Master Router
- Configuring the IP Interfaces
- Configuring the Master Router

Configuring the Protocol (IP) VLAN of the Master Router

```
------------------------------------------------------------
CB9000@slot3.1 [12-E/FEN-TX-L3] (): bridge vlan define
Enter VID (2-4094) [2]: 2
Select bridge ports (1-13|all|?): 13
Enter protocol suite

(IP,IPX,Apple,XNS,DECnet,SNA,Vines,X25,NetBEUI,unspecified,IPX-II,IPX-802.2,
IPX-802.3,IPX-802.2-SNAP): ip
Enter protocol suite ('q' to quit)
(IPX,Apple,XNS,DECnet,SNA,Vines,X25,NetBEUI,IPX-II,IPX-802.2,IPX-802.3,
IPX-802.2-SNAP): q
Configure layer 3 address? (n,y) [y]: n
Configure per-port tagging? (n,y) [y]: n
Enter VLAN Name {?} []: vlan-2

------------------------------------------------------------
CB9000@slot3.1 [12-E/FEN-TX-L3] (): bridge vlan summary all

VLAN summary

VLAN Mode: allOpen

VLAN-aware mode: allPorts

Index                              VID                 Type              Origin
1                                    1                  open               static
2                                    2                  open               static

Index                              Name                 Ports
1                              Default                 1-13
2                               vlan-2                   13
```

Configuring the IP Interfaces

```
------------------------------------------------------------
CB9000@slot3.1 [12-E/FEN-TX-L3] (): ip interface define
Enter IP address: 44.4.4.1
Enter subnet mask [255.0.0.0]: 255.255.255.0
Enter interface type (vlan,port) [vlan]: vlan
Enter VLAN interface index {2|?} [2]: 2

------------------------------------------------------------
CB9000@slot3.1 [12-E/FEN-TX-L3] (): ip int def 55.5.5.1 255.255.255.0 vlan
2

------------------------------------------------------------
CB9000@slot3.1 [12-E/FEN-TX-L3] (): ip interface summary all
IP routing is disabled

Index  IP address      Subnet mask      State  Type  ID
    1  44.4.4.1        255.255.255.0    Up     VLAN   2
    2  55.5.5.1        255.255.255.0    Up     VLAN   2
```

Configuring the Master Router

```
CB9000@slot3.1 [12-E/FEN-TX-L3] (): ip vrrp define
Enter virtual router's type (Primary,Backup) [Primary]: primary
Enter VLAN interface index {2|?} [2]: 2
Enter VRID (1-255) [1]: 52
Enter address mode (auto-learn,IP-address) [auto-learn]: auto
Enter primary IP Address index {1-2|?} [1]: 1
Enter the advertise interval in sec (1-255) [1]: 1
Enter Authentication Type (none,pass) [pass]: none

------------------------------------------------------------
CB9000@slot3.1 [12-E/FEN-TX-L3] (): ip vrrp mode
Enter VLAN interface index (2|?) [2]: 2
Enter virtual router ID (52|?) [52]: 52
Vrid 52 - Enter virtual router mode (enabled,disabled) [disabled]: enable

------------------------------------------------------------
CB9000@slot3.1 [12-E/FEN-TX-L3] (): ip vrrp detail all all

VLAN Index: 2  Ports: 13
VRID  State   Interval  Pri  Preempt  Mode    Auth  Password  AddrMode  Error
 52   Master  1 sec.    255  yes      enable  none  N/A       learn     none

      Address         PrimaryIpAddr    MasterIpAddr
      44.4.4.1        44.4.4.1         44.4.4.1
      55.5.5.1
```

Configuring Router 2 as the Backup Router

Configuring Router 2 as the Backup router, as shown in Figure 48, involves the following tasks:

- Configuring the Protocol (IP) VLAN of the Backup Router
- Configuring the IP Interfaces
- Configuring the Backup Router

Configuring the Protocol (IP) VLAN of the Backup Router

```
------------------------------------------------------------
CB9000@slot5.1 [10-E/FEN-FX-L3] (): bridge vlan define
Enter VID (2-4094) [2]: 2
Select bridge ports (1-11|all|?): 11
Enter protocol suite
 (IP,IPX,Apple,XNS,DECnet,SNA,Vines,X25,NetBEUI,unspecified,IPX-II,IPX-802.2,
IPX-802.3,IPX-802.2-SNAP): ip
Enter protocol suite ('q' to quit)
 (IPX,Apple,XNS,DECnet,SNA,Vines,X25,NetBEUI,IPX-II,IPX-802.2,IPX-802.3,
IPX-802.2-SNAP): q
Configure layer 3 address? (n,y) [y]: n
Configure per-port tagging? (n,y) [y]: n
Enter VLAN Name {?} []: vlan-2

------------------------------------------------------------
CB9000@slot5.1 [10-E/FEN-FX-L3] (): bridge vlan summary all

VLAN summary
VLAN Mode: allOpen

VLAN-aware mode: allPorts

Index                              VID                Type                Origin
1                                    1                open                static
2                                    2                open                static

Index                              Name               Ports
1                                Default              1-11
2                                 vlan-2                11
```

Configuring the IP Interfaces

```
------------------------------------------------------------
CB9000@slot5.1 [10-E/FEN-FX-L3] (): ip int define 44.4.4.2 255.255.255.0 vlan 2

CB9000@slot5.1 [10-E/FEN-FX-L3] (): ip int define 55.5.5.2 255.255.255.0 vlan 2

CB9000@slot5.1 [10-E/FEN-FX-L3] (): ip int summary all
IP routing is disabled

Index  IP address       Subnet mask      State  Type  ID
    1  44.4.4.2         255.255.255.0    Up     VLAN    2
    2  55.5.5.2         255.255.255.0    Up     VLAN    2
```

Configuring the Backup Router

```
-----------------------------------------------------------------------------

CB9000@slot5.1 [10-E/FEN-FX-L3] (): ip vrrp define
Enter virtual router's type (Primary,Backup) [Primary]: backup
Enter VLAN interface index {2|?} [2]: 2
Enter VRID (1-255) [1]: 52
Enter address mode (auto-learn,IP-address) [auto-learn]: auto
Enter primary IP Address index {1-2|?} [1]: 1
Enter backup virtual router priority (1-254) [100]: 100
Enter the advertise interval in sec (1-255) [1]: 1
Enter virtual router preempt mode (no,yes) [yes]: yes
Enter Authentication Type (none,pass) [pass]: none

------------------------------------------------------------
CB9000@slot5.1 [10-E/FEN-FX-L3] (): ip vrrp mode 2 52 enable

------------------------------------------------------------
CB9000@slot5.1 [10-E/FEN-FX-L3] (): ip vrrp detail all all

VLAN Index: 2  Ports: 11
VRID  State   Interval  Pri  Preempt  Mode     Auth  Password  AddrMode  Error
 52   Backup  1 sec.     100  yes      enable   none  N/A        learn     none

      Address          PrimaryIpAddr    MasterIpAddr
      44.4.4.1          44.4.4.2         44.4.4.1
      55.5.5.1

Virtual Router statistics:

                        becomeMaster      advertReceived         advIntErrors
                                   0                  35                    0
```

Switching from Master Router to Backup Router

To see how the Backup router assumes Master router responsibilities, disable the Master Router, then display the VRRP configuration. The display shows that the IP address of the Master has changed to that of the former backup virtual router.

Disabling the Master Router

```
--------------------------------------------------------------------------------

CB9000@slot3.1 [12-E/FEN-TX-L3] (): eth portstate 13 disable
```

Displaying the Results of the Master Router Change

```
CB9000@slot5.1 [10-E/FEN-FX-L3] (): ip vrrp detail all all

VLAN Index: 2  Ports: 11
VRID  State   Interval  Pri  Preempt  Mode     Auth  Password  AddrMode  Error
 52   Master  1 sec.    100  yes      enable   none  N/A       learn     none

      Address          PrimaryIpAddr    MasterIpAddr
      44.4.4.1         44.4.4.2         44.4.4.2
      55.5.5.1

Virtual Router statistics:

                         becomeMaster      advertReceived        advIntErrors
                                    1                  61                   0
```

Standards, Protocols, and Related Reading

Virtual Router Redundancy Protocol is defined in the IETF Request For Comments (RFC) document RFC2338. RFC2338 can be found at the following WWW site:

`http://www.ietf.cnri.reston.va.us/rfc/rfc2338.txt`

The Internet Assigned Numbers Authority (IANA), assigns and maintains lists of all assigned numbers used for operation of the Internet (protocol type, Ethernet codes, PPP codes, IP port numbers, ICMP parameters, IP Multicast addresses, HTTP parameters, IEEE 802 numbers, and so forth).

The Directory of General Assigned Numbers can be found at the following site:

`http://www.iana.org/numbers.html`

18 IP MULTICAST ROUTING

This chapter provides conceptual information, configuration options, and implementation guidelines for IP multicast routing on Switch 4007 Multilayer Switching Modules. This chapter covers these topics:

- IP Multicast Overview
- How a Network Supports IP Multicast
- Key Concepts
- How IGMP Supports IP Multicast
- How DVMRP Supports IP Multicast
- Key Guidelines for Implementation
- Configuring IGMP Options
- Configuring DVMRP Interfaces
- Configuring DVMRP Tunnels
- Configuring DVMRP Default Routes
- Viewing the DVMRP Routing Table
- Viewing the DVMRP Cache
- Using IP Multicast Traceroute
- Standards, Protocols, and Related Reading

You can manage IP multicast routing parameters from the `ip multicast` *menu in the Administration Console of Multilayer Switching Modules. See the* Switch 4007 Command Reference Guide*.*

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Switch 4007 Multilayer Switching Modules use two protocols to support IP multicast routing: the Internet Group Management Protocol (IGMP) and the Distance-Vector Multicast Routing Protocol (DVMRP). IGMP is also supported on Layer 2 Switching Modules, but that implementation is described in Chapter 11.

IP Multicast Overview

The easiest way to begin to understand multicasting is to compare it against two other address types and their communication models.

Unicast Model

A *unicast* address is designed to transmit a packet from a source to a single destination. Unicast transmissions are for *one-to-one* communication. If multiple users need to receive the same communication, the source operating in unicast mode generates and sends each copy separately.

Broadcast Model

A *broadcast* address is used to send a datagram from a source to multiple destinations — an entire subnetwork, for example. Broadcast transmissions produce *one-to-many* communication, but some of the receivers may not want or need to receive the communication.

Multicast Model

A *multicast* address is used for *one-to-many* and *many-to-many* communication in an environment where users and network devices either explicitly or implicitly communicate their desire to receive the communication.

In contrast to unicast, a source that uses IP multicast generates and sends only *one* copy of the information that is desired by multiple receivers. At point where the delivery path that reaches group members diverges, network devices replicate and forward the packets. This approach makes efficient use of both source processing power and network bandwidth.

When using the Internet Protocol (IP) as the basis for multicast communication, the requests for and delivery of the communication is fundamentally controlled by referencing certain IP addresses or their MAC-based equivalents. These addresses are called group addresses or *groups* and hosts that reference these addresses are called *group members.*

IP multicast group members can be scattered across multiple subnetworks; thus, successful transmission from a source to group members can occur within a campus LAN, a MAN, or over a WAN.

As an extension to the standard IP network-level protocol, IP multicast was first defined in 1985 in RFC 966. Certain other protocols are used to support IP multicast processes. These are explained later in this chapter.

Benefits of IP Multicast

New applications that are designed to increase productivity within and across organizations are driving the need for network infrastructures to support IP multicast. When the application content is time-sensitive or requires significant bandwidth (for example, a video stream), the IP multicast process provides an efficient delivery mechanism.

The business benefits of using IP multicast are that it:

- Enables the simultaneous delivery of information to many receivers in the most efficient, logical way.
- Vastly reduces the load on the source (for example, a server) because it does not have to produce multiple copies of the same data.
- Makes efficient use of network bandwidth and scales as the number of participants or collaborators expands.
- Works in concert with other protocols and services, such as Quality of Service (QoS) and Resource Reservation (RSVP) requests to support real-time multimedia.

How a Network Supports IP Multicast

To support IP multicast, the sending and receiving nodes, as well as the network infrastructure between them, must be multicast-enabled. Specifically, there must be cohesive support for IP multicast in the following components: TCP/IP protocol stack, operating systems, application software, NICs, and Layer 3 devices. Support for IP multicast in Layer 2 devices is not required by the standard, but is desirable, as explained later in this section.

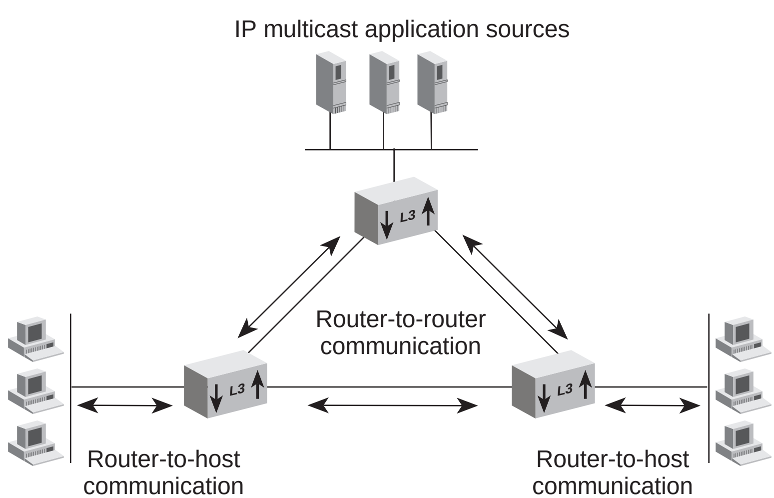
IP Multicast Routing

IP multicast transmissions fundamentally depend on multicast-enabled Layer 3 devices (traditional routers or Layer 3 switches; hereafter both are called *routers*) to direct packets on an efficient path from sources to destinations.

As shown in Figure 50, routers that support IP multicast must accomplish two important tasks:

- Communicate with other routers to determine the shortest, loopfree delivery path between an IP multicast source and its group members
- Communicate with hosts on its directly attached subnetworks to determine which hosts want to join or leave IP multicast groups

Figure 50 IP Multicast Communication Processes



Supporting Protocols in Your Module

To communicate with other routers, Switch 4007 Multilayer Switching Modules support the Distance-Vector Multicast Routing Protocol (DVMRP) version 3.6. DVMRP functions and configuration options are explained later in this chapter.

To communicate with group members on directly attached subnetworks, Switch 4007 Multilayer Switching Modules support the Internet Group Management Protocol (IGMP) version 1 and version 2. IGMP functions are covered later in this chapter.

IP Multicast Tunnels

IP multicast routers are key connection points for delivering IP multicast traffic between sources and multicast group members. In the event that some routers in your network only transmit unicast packets, you can configure a transitional technique called *tunneling* to extend the service area. Tunnels provide a virtual point-to-point link between two multicast routers, where the path between them includes one or more routers that do not support multicast routing (unicast routers).

Figure 51 DVMRP Tunnel Example

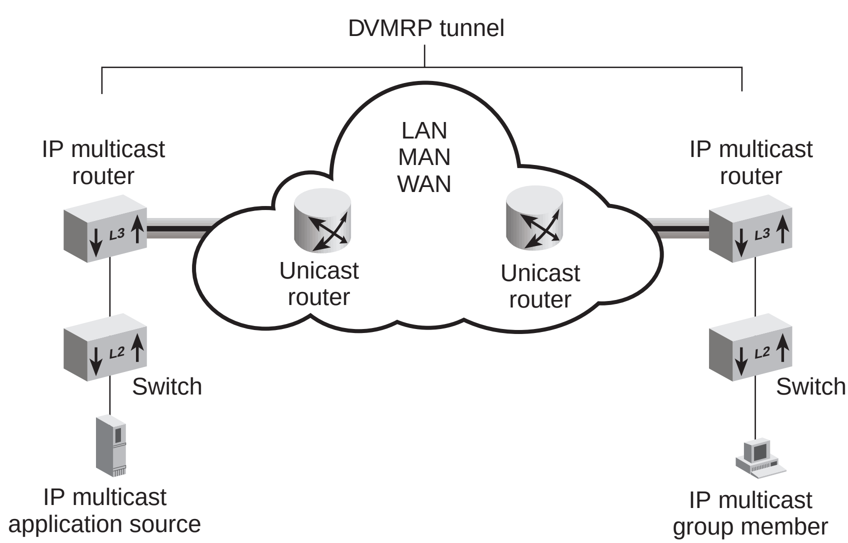


Figure 51 depicts a network configuration that requires an tunnel in order for the PC to receive the IP multicast application. The multicast routers support DVMRP, thus the tunnel is also configured with that protocol.

A multicast router is required at each end of the tunnel. At each tunnel entrance, the router encapsulates the IP multicast packets in standard IP unicast packets — that is, it puts them in a format that the unicast routers can understand. When these packets reach the end of the tunnel, the router strips the encapsulation away and returns the packet to its native IP multicast format.

Supporting Protocol in Your Module

Switch 4007 Multilayer Switching Modules use the Distance-Vector Multicast Routing Protocol (DVMRP) to form IP multicast tunnels. Specific aspects of tunnel configuration are described later in this chapter.

IP Multicast Filtering

When a router discovers that at least one IP multicast group member resides on a directly attached subnetwork, it forwards group traffic on that interface until it determines that group members no longer require the traffic. If multiple ports are configured in an interface, a router sends copies of the group traffic to all ports, even if only one port of those ports leads to group members. This is because the multicast routing protocol does not track *exactly* where group members reside on that interface.

The ability to filter IP multicast traffic on ports within a routing interface that do not lead to group members is highly desirable (although it is not required in the IP multicast standard) because it allows you to further optimize the LAN environment. Through targeted filtering, a router can conserve even more network bandwidth and minimize unnecessary interruptions to endstation CPUs.

It is also important to have a similar IP multicast filtering capability in Layer 2 switches. See Chapter 11 to learn about this capability in Switch 4007 Layer 2 Switching Modules.

Supporting Protocol in Your Multilayer Switching Module

Your module supports the Internet Group Management Protocol (IGMP) version 1 and version 2. to track exactly which ports in an interface require IP multicast group traffic. IGMP covers two main functions: querying and snooping. These are explained later in this chapter.

Internet Support for IP Multicast

The MBONE is the Internet's experimental multicast backbone network. It is an interconnected set of Internet routers, subnetworks, and tunnels that support the delivery of IP multicast traffic.

The MBONE was first configured in 1992 as a test zone to enable IP multicast applications to be deployed without waiting for multicast routers to replace unicast routers across the entire Internet. The MBONE is actually a virtual network located within portions of the physical Internet. Its construction reflects several multicast zones connected together via IP multicast tunnels. When it was created in 1992, the MBONE spanned four countries and 40 subnetworks; today it spans over 25 countries and thousands of subnetworks.

You can connect to the MBONE through most Internet service providers (ISPs). You can use it to test multicast applications and technology or to connect private multicast LANs. Some organizations broadcast public information over the MBONE; examples include IETF (Internet Engineering Task Force) meetings and NASA (National Aeronautics and Space Administration, United States) space shuttle launches.

Key Concepts

This section describes several terms and concepts related to IP multicast routing.

Traffic Movement

Application sources generate the majority of IP multicast packets, but group members and routers that are communicating (DVMRP and IGMP messages) to establish the delivery path also generate IP multicast packets.

Traffic from application sources always travels in one direction — *downstream* from the source to group members. Using various protocols, network devices are responsible for determining where group members exist and coordinating a loop-free delivery path from the source to them.

Traffic that relates to the delivery path can travel both *upstream* and *downstream* — between routers and between routers and group members.

IP Multicast Groups

Users can join or leave an IP multicast group at any time. Users request and cancel membership through mechanisms built into their desktop application — perhaps visible to the user as *Go* and *Quit* buttons. There are no restrictions on the physical location or number of members in a group. A user may belong to one or more groups at any time.

Source-Group Pairs

Each IP multicast transmission can be linked to a unique pairing of a source address and multicast group address (destination address). In addition, network devices form a unique delivery path for each source-group pair. Multicast routers and switches track information about each source-group pair — mainly, the location of group members — and dynamically adjust the delivery path to ensure that IP multicast packets are delivered only where they need to go.

Multicast Addresses

A multicast packet differs from a unicast packet by the presence of a *multicast group address* in the destination address field of the IP header. IP multicast uses a Class D destination address format, which has the high-order four bits set to *1-1-1-0* followed by a 28-bit multicast group identifier.

Registered Groups

The Internet Assigned Numbers Authority (IANA) maintains a list of registered IP multicast groups. Expressed in standard dotted decimal notation, group addresses range from 224.0.0.0 – 239.255.255.255 and are classified by IANA as follows:

- Addresses 224.0.0.0 – 224.0.0.225 are reserved for use by protocols and other special functions. See Table 78 for examples of permanent reserved addresses, or for a complete and current list, visit the IANA Web site:

  `http://www.iana.org`

- Addresses 224.0.1.0 – 239.255.255.255 are either assigned to various multicast applications or remain unassigned. From this range, addresses 239.0.0.0 – 239.255.255.255 are reserved for site-local applications, not Internet-wide applications.

Table 78 Examples of Class D Permanent Address Assignments

Address	Meaning
224.0.0.0	Base Address (Reserved)
224.0.0.1	All systems on this subnet
224.0.0.2	All routers on this subnet
224.0.0.4	All DVMRP routers
224.0.0.5	All OSPF routers
224.0.0.6	All OSPF designated routers
224.0.0.7	All ST routers
224.0.0.8	All ST hosts
224.0.0.9	All RIP version 2 routers
224.0.0.11	Mobile agents
224.0.0.12	DHCP server/relay agent
224.0.0.13	All PIM routers
224.0.0.14	RSVP, Encapsulation
224.0.0.15	All CBT routers

Reserved MAC Addresses

IANA also controls a reserved portion of the IEEE-802 MAC-layer multicast address space. All addresses in this block use hexadecimal format and begin with 01-00-5E. A simple procedure maps Class D addresses to this block, so that IP multicasting can take advantage of the hardware-level multicasting supported by network interface cards (NICs).

The mapping process involves placing the low-order 23 bits of the Class D address (binary format) into the low-order 23 bits of the MAC address (hexadecimal format). For example, the Layer 3 address 224.10.8.5 maps to the Layer 2 MAC address 01-00-5E-*0A-08-05*.

To send a multicast packet, a source station inserts the Class D address in the IP packet, the network interface card maps that address to a IEEE-802 Ethernet multicast MAC address, and sends the frame. A host that wants to receive packets that are addressed to this group notifies its IP layer as such.

How IGMP Supports IP Multicast

IGMP provides a way for routers and switches to learn where group members exist on a network, and thus provides a critical function in the IP multicast packet delivery process.

Electing the Querier

On each subnetwork or broadcast domain (VLAN), the communication between routers, switches, and group members begins with one IGMP-capable device being elected as the *querier* — that is, the device that asks all hosts to respond with a report of the IP multicast groups that they wish to join or to which they already belong. The querier is always the device with the lowest IP address in the subnetwork. It can be a router or a Layer 2 switch. The network traffic flows most efficiently if the querier is the closest device to the sources of IP multicast traffic.

Query Messages

The querier normally sends messages called *IGMP Host Membership Query Messages*, or *queries*, every 125 seconds. All the hosts hear the query because it is addressed to 224.0.0.1, the *all systems on this subnetwork* Class D address. A query is not forwarded beyond the subnetwork from which it originates.

Host Messages

Hosts use IGMP to build their own types of IP multicast messages, as described in this section.

Response to Queries

Hosts respond to queries with *IGMP Host Membership Report* messages, or simply *IGMP reports*. These reports do not travel beyond their origin subnetworks, and hosts send them at random intervals to prevent the querier from being overwhelmed.

A host sends a separate report for each group that it wants to join or to which it currently belongs. Hosts do not send reports if they are not group members.

If a router does not receive at least one host report for a particular group after two queries, the router assumes that members no longer exist and it prunes the interface for that source-group spanning tree.

Join Message

Rather than wait for a query, a host can also send an IGMP report on its own initiative to inform the querier that it wants to begin receiving a transmission for a specific group (perhaps by clicking a *Go* or *Start* button on the client interface). This is called a *join* message. The benefit is faster transmission linkages, especially if the host is the first group member on the subnetwork.

Leave-Group Messages

Leave-group messages are a type of host message defined in IGMP version 2. If a host wants to leave an IP multicast group, it issues a leave-group message addressed to 224.0.0.2, the *all routers in this subnetwork* Class D address. Upon receiving such a message, the querier determines whether that host is the last group member on the subnetwork by issuing a *group-specific query*.

Leave-group messages lower *leave latency* — that is, the time between when the last group member on a given subnetwork sends a report and when a router stops forwarding traffic for that group onto the subnetwork. This process conserves bandwidth. The alternative is for the router to wait for at least two queries to go unanswered before pruning that subnetwork from the delivery tree.

Role of IGMP in IP Multicast Filtering

To further refine the IP multicast delivery process and maximize bandwidth efficiency, a Layer 3 module filters IP multicast packets on appropriate ports using a process called *IGMP snooping*. Both bridged interfaces and routed interfaces record which ports receive host IGMP reports and then set their filters accordingly so that IP multicast traffic for particular groups is not forwarded on ports or VLANs that do not require it.

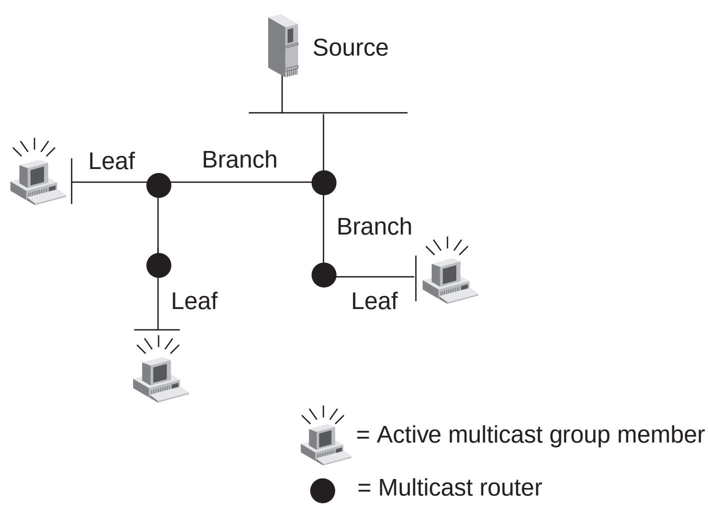
How DVMRP Supports IP Multicast

DVMRP is a distance-vector routing protocol that allows routers to establish shortest-path, source-rooted, IP multicast delivery trees. While it is similar to the Routing Information Protocol (RIP), one important difference is that DVMRP focuses on the *previous hop* back to a multicast source, not the next hop to a destination. Multicast routers are concerned with moving packets *away from the source* on a loopless path so that multicast storms do not occur.

Spanning Tree Delivery

DVMRP version 3.x uses the Reverse Path Multicast (RPM) algorithm to construct a delivery tree that begins at the source and spans out to reach group members on a loopless path through the network. Hence, DVMRP seeks to form a *source-rooted spanning tree* for each source-group pair. The shape of each tree changes dynamically, depending on the location of hosts that join and leave the group. As shown in Figure 52, any routing interface that contains group members is called a *leaf interface*.

Figure 52 Sample IP Multicast Spanning Tree



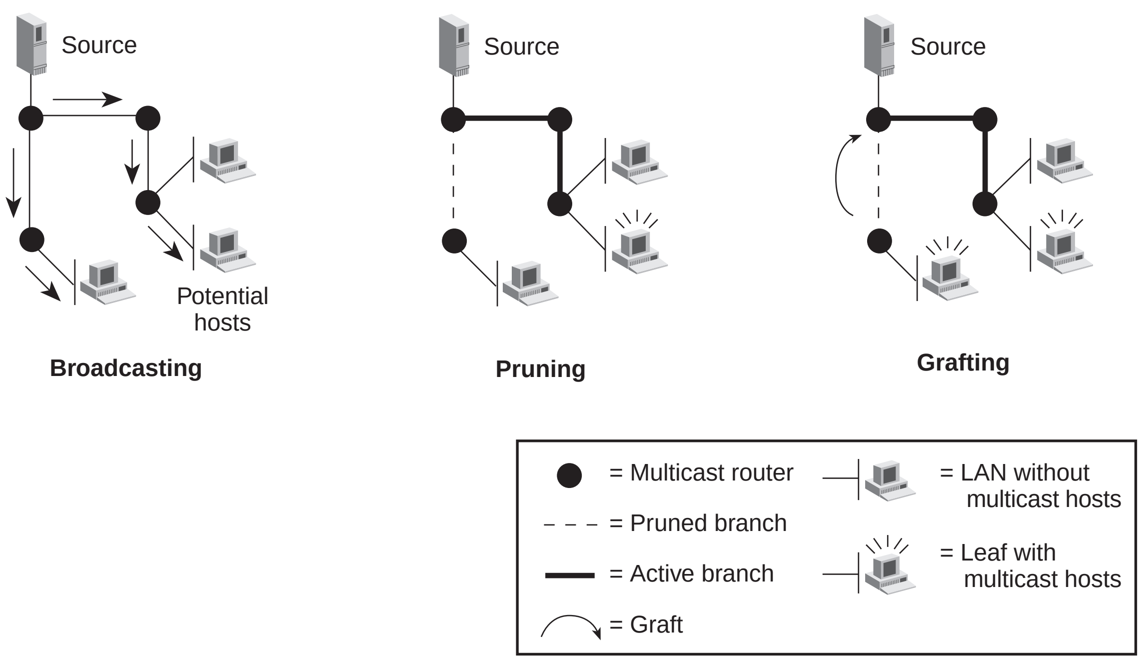
The term *spanning tree* applies to any loopless graph that spans intelligent nodes. The DVMRP spanning tree structure provides only one active path to connect any two multicast routers in the network. This approach provides a logical, efficient path to reach group members and prevents multicast storms from decreasing network performance.

The Spanning Tree Algorithm that is specified in the IEEE 802.1D MAC Bridges base standard is a different implementation of the spanning tree concept; it is not used with IP multicast.

Managing the Spanning Tree

RPM uses three main techniques to dynamically adjust the shape of an IP multicast spanning tree: broadcasting, pruning, and grafting. These techniques balance the goal of an efficient delivery path with the goal of effective service for all potential group members. Figure 53 shows the broadcasting, pruning, and grafting processes.

Figure 53 RPM Techniques for Managing the Multicast Spanning Tree



Interface Relationships

The interface on which a router receives source-origin traffic for a given source-group pair is called the incoming or *parent* interface. Each interface over which the router forwards source-group traffic is called an outgoing or *child* interface. A child interface on one router can:

- **Be a leaf interface** — A subnetwork with group members
- **Lead to the parent interface of a downstream router** — The next router in the delivery path to reach group members

Broadcasting

The first packet for any source-group pair is broadcast across the entire network, as far as packet time-to-live (TTL) and router TTL thresholds permit. If a packet arrives on an interface that the router determines to be the shortest path back to the source (by comparing interface metrics), then the router forwards the packet on all interfaces except the incoming interface. Downstream routers quickly send either:

- Prune messages (explained next) to upstream routers if their interfaces do not lead to group members
- IGMP reports if they want to continue receiving traffic for that source-group pair.

Some IP multicast applications try to actively send traffic on the network, even if no group members are requesting their traffic. Your module can detect which ports lead to routers and send these infrequent broadcast *packets only to those ports. Otherwise, the module filters all IP multicast group traffic for which it has received no IGMP Reports or graft messages.*

Pruning

A parent interface transmits *a prune* message to its upstream neighboring router if there are no group members on its child interfaces. A prune message directs the upstream router not to forward packets for a particular source-group pair in the future. Prune messages always affect the entire routing interface; they cannot be targeted to prune individual port segments that belong to an interface (IGMP snooping effectively achieves this, however).

Prune messages always begin at the leaf routers and are sent one hop back toward the source. Each successive router determines whether to prune its connections.

Inside the prune message is a prune *lifetime*, or prune *timer*, which is a period of time for which the prune message is valid. When the prune lifetime expires, the interface is added back into the multicast delivery tree — that is, until it generates another prune message.

Even though routers must use memory to store membership and prune information, this approach regains the bandwidth that would have been wasted on branches that do not lead to group members.

Grafting

If a router that has previously sent a prune message discovers a new group member (from IGMP Reports) on one of its connections, it sends a *graft* message to the previous hop router. When an upstream router receives this message, it cancels the prune message it previously received. Hop by hop, graft messages cascade back toward the source until they reach the nearest live branch point on the IP multicast spanning tree.

DVMRP Interface Characteristics

All DVMRP interfaces and DVMRP tunnels have two characteristics: a metric that specifies the *cost* for the interface and a time-to-live (TTL) threshold.

- **Metric Value** — The DVMRP metric is a numeric value or cost for that path. The higher the assigned cost, the less likely it is that the multicast packets will be routed over that interface (provided that other path options exist).
- **TTL Threshold** — Each IP multicast packet uses the TTL field of the IP header to provide a scope-limiting parameter for routers to work with. The initial value that the source sets in the TTL field controls the number of router hops that the IP multicast packet can make through the network. Each time that a router forwards a packet, it decrements the packet TTL by one. As long as the multicast packet TTL is greater than the TTL threshold of the multicast router interface, the router forwards the packet. If not, the router filters (drops) the packet.

In all cases where the multicast router drops multicast packets, the router does not provide an error notification to the source because IP multicast is a connection-less technology.

Key Guidelines for Implementation

A Switch 4007 Multilayer Switching Module needs to have IP multicast routing features enabled only if network users that sit downstream of the module (from the perspective of the source location) require access to IP multicast application traffic.

Configuration Procedure

To activate IP multicast routing and filtering capabilities in a Multilayer Switching Module, follow this general procedure:

1. Configure VLANs and IP routing interfaces on the module. See Chapter 14 for more information about VLANs. See Chapter 16 for more information about IP routing.

2. Ensure that IGMP snooping and querying functions are enabled on the module.

   For general information about IGMP, see "How IGMP Supports IP Multicast" earlier in this chapter.

   For information about configuring IGMP functions on a Layer 3 module, see "Configuring IGMP Options" later in this chapter.

   To figure IGMP functions on Switch 4007 Layer 2 Switching Modules, see Chapter 11 in this guide.

3. Enable DVMRP on each interface that is to perform IP multicast routing.

   You can modify the default TTL threshold and DVMRP metric values for each interface.

   For general information about DVMRP see "How DVMRP Supports IP Multicast" earlier in this chapter.

   For information about configuring DVMRP, see "Configuring DVMRP Interfaces" later in this chapter.

4. If your network requires a multicast tunnel and if a Multilayer Switching Module is to going to serve as one or more tunnel endpoints, configure the tunnels now. (Remember to configure the tunnels on the remote systems as well.)

   See "Configuring DVMRP Tunnels" later in this chapter.

5. Configure a default route on an interface (if applicable to your network).

   See "Configuring DVMRP Default Routes" later in this chapter.

6 View the various displays, routing table and cache to see how the module is processing IP multicast traffic.

See "Viewing the DVMRP Routing Table" and "Viewing the DVMRP Cache" later in this chapter.

7 Use the traceroute option for troubleshooting or to determine the traffic paths.

See "Using IP Multicast Traceroute" later in this chapter.

Impact of Multicast Limits

As described in Chapter 9, you can use the `bridge port multicastLimit` option to set per-port multicast limits. This *optional* feature can prevent segments of your network from being adversely affected by multicast or broadcast storms. If network users have trouble receiving IP multicast application traffic, verify that bridge ports are not configured with a bridge port multicast limit that is too low.

Impact of IEEE 802.1Q on Multicasts

Multicasting in 802.1Q VLAN tagging environments may have performance implications for a Multilayer Switching Module. Specifically, if you have multiple VLANs associated with a single port, the module is forced to replicate multicast packets to each VLAN that has multicast group members, even if the path to reach the members is the same physical link. To achieve wirespeed multicast performance, 3Com recommends that you configure only one VLAN per port. Contact your 3Com representative about network design options.

Protocol Interoperability

Routing protocols other than DVMRP exist to support IP multicast functions. Interoperability issues between these routing protocols mean that you should plan your routing infrastructure carefully. Contact your 3Com representative for more information.

Configuring IGMP Options

You can enable or disable IGMP snooping and querying functions, set the interface time-to-live (TTL) threshold, and obtain summary and detail displays of IGMP-related information.

Querying and Snooping Modes

Your Multilayer Switching Module divides IGMP functions into two modes:

- **Querying** — Allows an IP multicast routing interface to function as the querier if so elected.
- **Snooping** — Allows the module to forward multicast packets only to the appropriate ports within its routing or bridging interfaces.

Important Considerations

- Both modes are enabled as the factory default. These settings apply to the entire module. You cannot enable or disable snooping or querying on specific interfaces.
- 3Com recommends that you keep both modes enabled at all times. They add little processing overhead to the module.

Configuring DVMRP Interfaces

DVMRP is the protocol used to develop source-rooted spanning trees between routers in the network. You can enable or disable DVMRP on individual routing interfaces.

Important Considerations

- The default setting for DVMRP on each new interface is disabled.
- If DVMRP is disabled, the interface cannot participate in forming IP multicast spanning trees. If the Internet Group Management Protocol (IGMP) is enabled, the module can still forward IP multicast traffic.
- Enabling DVMRP causes the module to assign default values of `1` for both the TTL threshold and metric on the interface. You can modify these values at any time.
- A TTL threshold value of `1` means the interface forwards all IP multicast packets except those which have expired (packet TTL is 0). Before you change the TTL threshold value, consider the relative location of the module in the network and your networking objectives.

Table 79 lists conventional numeric values and network objectives.

Table 79 Conventional TTL Scope Control Values

TTL Value	Objective
0	Restricted to the same host
1	Restricted to the same subnetwork
16	Restricted to the same site
64	Restricted to the same region
128	Restricted to the same continent
255	Unrestricted in scope

Configuring DVMRP Tunnels

A DVMRP tunnel allows IP multicast packets to traverse a portion of your network infrastructure that is not multicast-aware. In Multilayer Switching Modules, you can define tunnels, modify tunnel characteristics, display information about tunnels you have defined, and remove tunnels.

Important Considerations

- All networks do not require DVMRP tunnels. A network needs a tunnel only if IP multicast packets must go through one or more unicast routers to reach IP multicast group members.
- You can configure any routing interface on a Multilayer Switching Module to be a DVMRP tunnel end point. The other tunnel end point must be a multicast interface on a different system and subnetwork.
- Before you can define a tunnel end point, you must configure a routing interface and enable DVMRP on the interface. Think of a tunnel end point as being layered on top of an existing IP multicast routing interface.
- The maximum number of IP multicast tunnels that you can define on a Switch 4007 Multilayer Switching Module is 8.
- To define a tunnel, you specify the following tunnel characteristics:
  - The index number of the local router interface that serves as the tunnel end point.
  - The IP address of the destination multicast router. *This address must be a remote address*. The destination multicast router cannot be directly connected to the same subnetwork.
  - When you define a tunnel, the module assigns tunnel metric and tunnel TTL threshold values of 1. You can modify these at any time.

- You must define the tunnel on both end points — that is, on both the local module and the remote system — even though you specify the address of the remote router interface in the local module.
- DVMRP interfaces and tunnels have similar characteristics (metric and TTL threshold), but the tunnel characteristics do not have to match the interface characteristics.
- If you try to remove an IP interface, and you have a tunnel defined on that interface, the module warns you with an error message. You must remove the tunnel before you can remove the IP interface.
- You can define multiple multicast tunnel end points on the same local routing interface, but each must lead to different remote end points.
- When you define a tunnel, the module assigns a tunnel index number to it. The multicast tunnel display lists tunnels in ascending order by the tunnel index number. Tunnel index numbers provide a way to identify and remove individual tunnels, which is especially useful when multiple tunnel end points are configured on the same routing interface.
- When you remove a tunnel, the module does not dynamically reorder remaining tunnels in the multicast tunnel display. For example, if you had three tunnels with tunnel index numbers 1, 2, and 3 and you then removed tunnel 2, the multicast tunnel display lists the remaining tunnels and identifies them with their original tunnel index numbers (1 and 3). The module does not dynamically reassign tunnel index numbers (does not change 3 to 2). In this example, the module assigns tunnel index 2 to the next *new* tunnel that you define. After the modules uses index 2, it can assign index 4 to the next new tunnel, and so on.
- Removing a tunnel end point on one system destroys that tunnel's functionality, but 3Com recommends that you remove the tunnel configuration from both systems.

Configuring DVMRP Default Routes

You can configure a default route for IP multicast traffic on any DVMRP routing interface in the module.

How Default Routes Work

If an interface is configured as a default route, it advertises source 0.0.0.0 to neighboring DVMRP routers. In their DVMRP routing tables, these neighboring routers list 0.0.0.0 as a source and list the advertising router interface as the *gateway* to reach that source.

Thus, if a neighboring router receives an IP multicast packet for which it has no normal routing information in its routing table, instead of filtering the packet, the router forwards it to the router which advertises the default route.

How to Configure A Default Route

To configure a default route on an interface, you

- Specify the interface index number.
- Set the default route metric.

  Specify a value from 1 through 32 to signify the cost of the route.
- Set the default route mode.

  There are two options:

  - **all** — The interface advertises the default route plus all other known routes to neighboring DVMRP routers.
  - **only** — The interface advertises only the default route to neighboring DVMRP routers.

Important Considerations

- If the module learns a default route, it propagates it no matter which mode is set on a given interface.
- The module allows you to configure an interface as a DVMRP default route, even when DVMRP is disabled on the interface. If DVMRP is disabled, the interface does not advertise itself as a default route.

Viewing the DVMRP Routing Table

Your module records DVMRP route information in a table that you can access from the management interface. Your module learns source-based route information from neighboring DVMRP routers and also advertises routes that it learns to its neighbors. The routing table does not consider group membership or prune messages. It simply records path information it has learned on its own or from other routers, including:

- Subnetworks from which IP multicast traffic originates
- Upstream routers (gateway) from which the module should expect to receive traffic from origin subnetworks
- Index number of the interface (parent) that is connected to the upstream router
- Outgoing interfaces (children) on which it could forward traffic if group members exist.

The module may never actually process IP multicast traffic from the sources listed in the routing table. This depends on whether group members exist on directly-attached subnetworks or on subnetworks from downstream routers.

See the *Command Reference Guide* for definitions of the fields of information and symbols used in the DVMRP route display.

Viewing the DVMRP Cache

Your module records information about the IP multicast group traffic it has processed. You can see this information in the DVMRP cache.

To display the DVMRP cache, the module prompts you to enter:

- A multicast source address
- A multicast group address

This process limits the cache table to displaying information for one source-group pair at a time. To display cache information for all source-group pairs, enter `255.255.255.255` at both prompts.

See the *Command Reference Guide* for definitions of the fields of information and symbols used in cache display.

Using IP Multicast Traceroute

You can perform an IP multicast traceroute from a Layer 3 module. The ability to trace the path of a IP multicast group packet from a source to a particular destination is desirable for troubleshooting purposes.

Unlike unicast traceroute, IP multicast traceroute requires the ability for routers to understand a special IGMP packet type and the related processes.

Beginning a trace from an IP multicast source would be difficult because, at forks in the network paths, there is no way to determine which direction to take. You would have to flood the entire tree and wait for responses (or the lack thereof) to find the path. Thus, a more efficient approach is to start at the destination and travel backwards toward the source, using the knowledge held by IP multicast routing protocols that work by calculating previous hops back toward sources.

An IP multicast traceroute proceeds as follows:

1. At the destination node (your module), you specify a source and group address.
2. The module sends a traceroute Query packet to the last-hop multicast router (the upstream router for this source-group pair).
3. The last-hop router turns the Query packet into a Request packet by adding a response data block containing its interface addresses and packet statistics. It then forwards the Request packet via unicast to the router that it believes is the previous hop for the given source-group pair.
4. Each previous hop router adds its response data to the end of the Request packet, then forwards it via unicast to the next previous hop router.
5. Finally, the first-hop router — that is, the router that believes that the source-group packets originate on one of its directly-attached subnetworks — adds its data, changes the Request packet to a Response packet, and sends the completed response back to the destination node that issued the traceroute query.
6. You see a display that shows IP addresses of the interfaces that span from your module back to the source that you specify. The display also shows the number of hops back to those interfaces, the multicast routing protocols used, and the amount of time it takes to reach each hop from the receiver.

Important Considerations

- When using IP multicast traceroute, the module assumes that it is the destination for the source-group traffic. You cannot enter a different destination address.
- A Response packet may be returned to your module before reaching the first-hop router if a fatal error condition such as "no route" is encountered along the path. All interim devices must support IP multicast traceroute for you to see a complete path on the display.

Standards, Protocols, and Related Reading

DVMRP was first defined in RFC 1075 and has been modified in various Internet drafts. IGMP was first defined in RFC 1112 and has been modified in various Internet drafts. To learn more about DVMRP and IGMP, IP multicast technology, or related events, consult the following Web resources:

- **`http://www.3com.com`**
- **`http://www.ipmulticast.com`**
- **`http://www.ietf.org`**
- **`http://www.stardust.com`**

19 OPEN SHORTEST PATH FIRST (OSPF) ROUTING

This chapter provides guidelines and other key information about how to configure Open Shortest Path First (OSPF) on a Multilayer Switching Module. This chapter covers these topics:

- OSPF Overview
- Key Concepts
- Key Guidelines for Implementing OSPF
- Autonomous System Boundary Routers
- Areas
- Default Route Metric
- OSPF Interfaces
- Link State Databases
- Neighbors
- Router IDs
- OSPF Memory Partition
- Stub Default Metrics
- Virtual Links
- OSPF Routing Policies
- OSPF Statistics

After you log in to the system and connect to a slot that houses a Multilayer Switching Module, you can manage OSPF routing from the `ip ospf` *menu of the Administration Console. See the* Switch 4007 Command Reference Guide.

OSPF Overview

The OSPF link-state protocol dynamically responds to changes in network topology that occur within a group of networks and routers known as an *autonomous system*. OSPF tracks the states of links and routers in each autonomous system, and when a change occurs, calculates new routes based on the new topology. The OSPF protocol responds to network topology changes with a minimum of administrator involvement and routing traffic.

All OSPF routers within an autonomous system build and synchronize databases of the autonomous system's network topology. Using its database, each router calculates the shortest path trees to every destination within the autonomous system. With this dynamic table of shortest paths, OSPF converges on an optimum route faster than other routing algorithms, such as the Routing Information Protocol (RIP).

Routers that use a distance-vector protocol like RIP periodically exchange all or a portion of their tables, but only with their neighbors. Routers using a link-state protocol like OSPF send small portions of their tables throughout the network by flooding.

For information about how to perform IP routing, see Chapter 16.

Features

Your system supports OSPF Version 2 as defined in RFC 1583. OSPF routing on your system includes these features:

- **Areas** — You can subdivide an autonomous system (AS) into more manageable contiguous networks called areas. Areas increase stability, conserve router resources, and support route summarization — the consolidation of network addresses. For more information, see "Areas" later in this chapter.
- **Default route metric** — You can configure a router to advertise itself as the default router for the area, and you can specify a cost to be advertised with the default route. When area routers fail to find a specific match for a packet's destination, the router then forwards the packet to the default router, which then forwards the packet to the most logical destination. For more information, see "Default Route Metric" later in this chapter.

- **OSPF interfaces** — An OSPF interface is an IP interface that you configure to send and receive OSPF traffic. When you configure an OSPF interface, you define the behavior and role of the interface within the OSPF routing domain. For example, router priority determines designated router selection, cost determines the expense associated with using the interface, and the Hello interval directly affects how fast topological changes are detected. For more information, see "OSPF Interfaces" later in this chapter.
- **Link state databases** — OSPF routers advertise routes using link state advertisements. The link state database contains the link state advertisements from throughout the area to which an OSPF interface is attached. For more information, see "Link State Databases" later in this chapter.
- **Neighbors** — OSPF interfaces attached to a common network are called neighbors; adjacent neighbors exchange link state database information. On broadcast networks, neighbors are discovered dynamically using the Hello protocol. On nonbroadcast multiaccess networks, you must statically configure neighbors. Your system allows you to display all neighbors in the locality of the router, as well configure them when needed. For more information, see "Neighbors" later in this chapter.
- **Router IDs** — A router ID identifies the router to other routers within the autonomous system. In addition, it serves as a tie-breaker in the designated router election. You systems gives you three methods by which you can configure a router ID for an OSPF interface. For more information, see "Router IDs" later in this chapter.
- **OSPF memory partition** — You can display how much memory your system allocates for OSPF data processing and adjust this memory allocation if needed. For more information, see "OSPF Memory Partition" later in this chapter.
- **Stub default metrics** — External link state advertisements are not propagated into stub areas. Instead, the area border router for a stub area injects a single external default route into the area. Your system allows you to configure an area border router to advertise a single external default route into the stub area while specifying the cost of the default route. For more information, see "Stub Default Metrics" later in this chapter.

- **Virtual links** — All areas of an OSPF routing domain must connect to the backbone area. In cases where an area does not have direct, physical access to the backbone, you can configure a logical connection to the backbone, called a *virtual link*. Virtual links can also add fault-tolerance and redundancy to the backbone. For more information, see "Virtual Links" later in this chapter.
- **OSPF routing policies** — Routing policies let you control what external routes OSPF routers store in their routing tables, as well as what external routes they advertise. Although routing policies are not part of the OSPF protocol itself, you can use them for increased security, enhanced performance, and overall control of OSPF routing data. For more information, see "OSPF Routing Policies" later in this chapter.
- **OSPF statistics** — You can also display general statistics for specific OSPF interfaces. These statistics can give you an overview of OSPF activity on the interface. For more information, see "OSPF Statistics" later in this chapter.

Benefits

The benefits of OSPF are what set it apart from both RIP and other Shortest Path First-based algorithms before it. While designing OSPF, the Internet Engineering Task Force (IETF) proposed a number of modifications which dealt with improving the existing SPF model. These modifications ranged from improving fault-tolerance to reducing routing traffic overhead. This focus toward improving the existing SPF model resulted in the following OSPF capabilities:

- **No hop count limitation** — OSPF places no limit on hop count. This capability is extremely important in larger networks. For example, a RIP network that spans more than 15 hops (15 routers) is considered unreachable. With OSPF, hop count is no longer an issue.
- **Efficient use of bandwidth for router updates** — OSPF uses IP multicast to send link-state updates only when routing changes have occurred, or once every 30 minutes. RIP, on the other hand, uses a 30 second interval. This policy ensures less processing on routers that are not listening to OSPF packets and better use of bandwidth.

- **Ability to partition the network into more manageable *areas*** — Many autonomous systems in the Internet are large and complicated to manage. OSPF allows them to be subdivided into smaller, more manageable networks or sets of contiguous networks called *areas*. You can think of an area as a generalization of an IP subnetworked network. The topology of an area is hidden from the rest of the AS, which significantly reduces routing traffic and also serves to lend the area protection from bad routing data. By partitioning the network into areas, OSPF limits the topology map required in each router. This limitation in turn conserves processing and memory requirements in each router, as well as reduces the amount of link state information being flooded onto the network.
- **Authentication for protocol exchanges** — All OSPF protocol exchanges are authenticated, which means that only known, trusted routers can participate in routing updates. OSPF supports a variety of authentication schemes, with a single scheme configured for each area. This partitioning allows some areas to use much stricter authentication than others.
- **Host-specific and network-specific route support** — OSPF supports traffic forwarding to single hosts or networks. Each network the router knows has both an IP destination address and a mask. The mask indicates the number of nodes on the network. A mask of all ones (0xffffffff) indicates a presence of a single node on the network (called a *stub network*).
- **Support for designated and back-up designated routers** — OSPF works by exchanging information between *adjacent* routers, not *neighboring* routers. To avoid the need for every router on a LAN or area to talk to every other router on a multiaccess network (a network that has at least two attached routers), one router is elected as the designated router. The designated router is considered adjacent to all other routers in the area and exchanges information with them. Routers that are not adjacent to each other do not exchange information. Therefore, instead of all routers on the network sending Link State Advertisements (LSAs), only the designated router sends LSAs. This feature significantly reduces data and routing traffic.

- **Support for virtual links to noncontiguous areas** — As discussed earlier, OSPF can partition large autonomous systems into smaller, more manageable subdivisions, called areas. An OSPF backbone is responsible for distributing routing information between the areas of an autonomous system. This backbone itself has all the properties of an area and consists of those networks that are not contained in any area. Although the backbone must be contiguous, backbone routers can also be connected by means of a virtual link. Virtual links can be configured between any two backbone routers that have an interface to a common nonbackbone area. OSPF treats two routers that are joined by a virtual link as if they were connected by an unnumbered point-to-point network. For more information, see "Virtual Links" later in this chapter.
- **Variable length subnet mask support** — OSPF considers both the IP address and subnet mask in determining the best route for a packet. An IP address mask is distributed with each advertised route. The mask indicates the range of addresses that are being described by the particular route. Including this mask enables the implementation of variable-length subnet masks (VLSMs), which means that a single IP network number can be *subnetworked* or broken up into many subnetworks of various sizes. When networks are subnetworked, OSPF forwards each IP packet to the network that is the best match for the packet's destination. It determines the best match by examining both the network address and the mask for each destination and finding the longest or most specific match. VLSM support is a key advantage for OSPF, especially when you consider the shortage of IP addresses. Another advantage is its flexibility.
- **Ability to import non-OSPF routing information** — Connectivity from one autonomous system to another is achieved through OSPF autonomous system boundary routers (ASBRs). ASBRs can import external link advertisements that contain information about external networks from other protocols like RIP and redistribute them as LSAs to the OSPF network. In this way, ASBRs flood information about external networks to routers within the OSPF network.
- **Assurance of loop-free networks** — The algorithm that dynamically calculates routing paths within the autonomous system does not generate a path with internal loops.

Key Concepts

Before you configure OSPF on your system, review the following key concepts and terms discussed in these sections:

- Autonomous Systems
- Areas
- Neighbors and Adjacency
- Router Types
- Protocol Packets
- How OSPF Routing Works

Autonomous Systems

An *autonomous system* consists of a set of OSPF routers that exchange routing information. The network shown in Figure 54 later in this chapter contains two autonomous systems.

Using identical topology databases, each router in an autonomous system calculates shortest-path routes from itself to every known destination in the autonomous system. The routers create their topology databases using the data in link state advertisements (LSAs) from other routers in the autonomous system.

Areas

Autonomous systems can be subdivided into smaller, more manageable, groups of contiguous networks called *areas*. Each OSPF router in an area must have identical topological link state databases. These databases may include area links, summarized links, and external links that depict the topology of the autonomous system.

Neighbors and Adjacency

Instead of each router sending routing information to every other router on the network, OSPF routers establish adjacencies among neighboring routers. Only adjacent routers exchange routing information. This information is exchanged using Database Description packets, which are used to describe the contents of each router’s link state database.

Router Types

OSPF routers serve several different, often overlapping, functions:

- **Internal routers** — Internal routers connect only to networks that belong to the same area. An internal router runs one copy of the OSPF algorithm and maintains routing data only for its area.

  In Figure 54, backbone area 0 and routers 1, 2, 3, and 4 are internal routers. In area 1, routers 5 and 6 are internal routers.

- **Backbone routers** — Backbone routers have an interface to the backbone area. Area border routers are always backbone routers because you must configure them as being within the backbone area or connected to it by a virtual link.

  In Figure 54, routers 1, 2, 3, and 4, and area border routers 1, 2, 3, and 4 are all backbone routers.

- **Area border routers (ABRs)** — Area border routers connect directly to networks in two or more areas. An area border router runs a separate copy of the OSPF algorithm and maintains separate routing data for each area that is connected to it (including the backbone area). Area border routers also send configuration summaries for their attached areas to the backbone area, which then distributes this information to other OSPF areas in the autonomous system.

  In Figure 54, four area border routers link the areas in autonomous system A.

- **Autonomous system boundary routers (ASBRs)** — Autonomous system boundary routers exchange their autonomous system topology data with boundary routers in other autonomous systems. Every router inside an autonomous system knows how to reach the boundary routers for its autonomous system.

  In Figure 54, two ASBRs control traffic between two autonomous systems.

- **Designated routers (DRs)** — Designated routers advertise network link states for attached network segments. A link state advertisement lists all routers that are connected to a segment.

  The DR is considered adjacent to all routers in its area. As a result, the DR exchanges routing data with all routers that are connected to its network segment.

- **Backup designated routers (BDRs)** — Backup designated routers are given a lower priority value than the DR. They take over DR functions if the DR fails.

Router IDs

The OSPF router ID identifies a router to other routers within an autonomous system. OSPF uses three types of router identifiers, which take the form of an IP address:

- **Default** — An arbitrary ID that the system generates and uses as the default router ID
- **Interface** — The address of an IP interface on the router
- **Address** — An arbitrary user-defined ID in the form of an IP address

You cannot set the router id to either 0.0.0.0 or 255.255.255.255.

Protocol Packets

The OSPF protocol uses these types of packets:

- **Hello packets** — Router interfaces periodically transmit hello packets to identify and maintain communications with their neighbors. In nonmulticast networks, routers find neighbors by sending unicast hello packets to other statically configured routers.
- **Database description packets** — Neighbor routers use database description packets to synchronize their link state summary databases.
- **Link state request packets** — To collect network topology data, routers transmit link state request packets to their neighbors on the segment.
- **Link state update packets** — On receiving a link state request packet, a router floods packets containing its LSA data into the area or autonomous system that it serves. The information contained in the packets depends on the router's location and function in the network.
- **Link state ack(nowledge) packets** — Routers use these packets to acknowledge receipt of link state update packets.

How OSPF Routing Works

This section summarizes how the OSPF algorithm works for a router that meets these characteristics:

- Lies within an autonomous system area (an interior router)
- Is attached to a multiaccess network
- Is configured to be the designated router for its network

Starting Up

When the router starts, the interfaces that are configured to run OSPF begin in the *down* state. When the lower-level IP protocols indicate that an interface is available, the interface moves to the *waiting* state. It remains in this state until the designated router and backup designated router are chosen.

Finding Neighbors

The router sends out hello packets to locate its network neighbors. These packets also list the routers from which the sending router has *received* hello packets. When a router detects its own address in another router's hello packet, the two routers establish two-way communications as neighbors.

Establishing Adjacencies

If neighboring OSPF routers succeed in exchanging and synchronizing their link state databases, they appear as *adjacent* in all router and network link advertisements.

Electing the Backup Designated Router

OSPF selects a backup designated router for the network segment. This router takes over as the designated router if the current designated router fails.

The OSPF algorithm first eliminates all routers that have an assigned priority of *0*. OSPF then selects the backup designated router from among the routers on the segment that have *not* declared themselves to be the designated router (based on their configuration settings). If some routers have already declared themselves to be the backup designated router, OSPF limits the selection to one of them.

OSPF selects the candidate router with the highest priority. If candidate routers have the same priority, OSPF selects the router that has the highest router ID.

Electing the Designated Router

OSPF selects a designated router, which originates LSAs on behalf of the network segment. These advertisements list all routers (including the designated router) that are attached to the segment. The designated router also floods LSA packets throughout the segment to allow its neighbors to update their databases.

The OSPF algorithm first eliminates all routers that have an assigned priority of *0*. OSPF then selects a designated router from among the routers that have declared themselves to be the designated router (based on their configuration settings). If no routers have declared their candidacy, the backup designated router becomes the designated router, and OSPF selects a new backup designated router.

OSPF selects the candidate router with the highest priority. If candidate routers have the same priority, OSPF selects the router that has the highest router ID.

The designated router then becomes adjacent to all other routers on the network segment by sending Hello packets to them.

Calculating Shortest Path Trees

OSPF routers collect raw topological data from the LSAs that they receive. Each router then prunes this data down to a tree of the shortest network paths centered on itself. In a series of iterations, the router examines the total cost to reach each router or network node in its domain. By discarding all but the lowest-cost path to each destination, the router builds a shortest path tree to each destination, which it uses until the network topology changes.

Routing Packets

A packet's source and destination determine the routers that move it:

- **Intraarea** — When a packet's source and destination are in the same area, the packet is routed using internal router databases. No routers are used outside the area.
- **Interarea** — When a packet's source and destination are in different areas, the topology databases in the backbone area dictate the paths that are taken between areas.

You can use virtual links to influence the routes that are taken for interarea traffic. See "Virtual Links" later in this chapter.

- **To a stub area** — When a packet's destination is in a stub area (an area that does not accept external route advertisements), OSPF uses the area's predefined default route. You configure default routing in area border routers that serve an OSPF stub area, such as area border router 1 in Figure 54. For more information, see "Stub Default Metrics" later in this chapter.
- **To a different autonomous system** — When a packet's source and destination are in different autonomous systems, ASBRs compute the routing paths using data obtained from another protocol, such as the Border Gateway Protocol. The boundary routers flood these external routes throughout all areas in the autonomous system except stub areas.

Key Guidelines for Implementing OSPF

Consider the following guidelines when you design a scalable and dependable OSPF internetwork:

These parameters must be consistent across all routers

The following OSPF interface parameters must be consistent across all routers on an attached network:

- Hello interval
- Dead interval
- Password

Addressing scheme

The addressing structure that you implement can affect both the scalability and the performance of your OSPF internetwork. Consider the following guidelines when you define an addressing structure to use for your OSPF internetwork:

- Make the range of subnets that are assigned within each OSPF area should be contiguous to allow optimal summarization by area border routers (ABRs).
- Define the address space so that you can easily add new areas, restructure existing ones, or add additional routers as your network grows.

Router placement and participation

When you populate an area with OSPF routers, consider the following guidelines:

- Because OSPF uses a CPU-intensive algorithm, keep the maximum number of routers participating in OSPF exchanges in any given area to around 50. This number decreases the likelihood of performance problems that may be associated with router recalculation. If the link is of high quality and the number of routes is minimal, you can increase the number of area routers.
- Keep the maximum number of neighbors for any one router to around 60. Each time that a topological change occurs, a router exchanges information only with those neighbors with which it has formed an adjacency. On a multiaccess network, this neighbor count is only of concern to the designated and backup-designated router of an area because, on a multiaccess network, area routers do not exchange link-state information with each other. Instead, they exchange link-state information with only the designated and backup designated routers.

Autonomous System Boundary Routers

Autonomous system boundary routers (ASBRs) are the links between the OSPF autonomous system and the outside network. They exchange their autonomous system topology data with boundary routers in other autonomous systems.

ASBRs can import external link advertisements that contain information about external networks from other protocols like RIP and redistribute them as LSAs to the OSPF network. In this way, ASBRs flood information about external networks to routers within the OSPF network.

Every router inside an autonomous system knows how to reach the boundary routers for its autonomous system.

In Figure 54, two ASBRs control traffic between two autonomous systems.

Configuring an ASBR

A router becomes an ASBR as a by-product of other settings. A router becomes an ASBR if any operational in-band IP interface on the router:

- Has both OSPF and RIP disabled on that interface. Or,
- Has RIP configured as learn, advertise, or enabled on that interface.

The ASBR then generates external link state advertisements for these IP interfaces.

A router also becomes an ASBR if you have configured either of the following on the box:

- A default route metric
- Any static routes, including configuring a default route

A router *never* becomes an ASBR if all of the router's interfaces reside in a stub area.

This last rule overrides all other cases where a router can become an ASBR.

You create IP interfaces with the `ip interface` option.

You configure RIP on IP interfaces with the `ip rip` options.

You configure OSPF on IP interfaces with the `ip ospf` options.

You create default route metrics with the `ip ospf defaultRouteMetric define` option.

You create static routes with the `ip ospf policy` options.

Areas

To reduce the amount of routing information that travels through a network, and the corresponding size of the OSPF routers' topology databases, subdivide OSPF autonomous systems into *areas*. Each area has the following configurable parameters:

- **Area ID** — A 32 bit number that identifies the area to the OSPF autonomous system. The Area ID is specified in the form n.n.n.n, where $0 \leq n \leq 255$. Subdividing the autonomous system (AS) into multiple areas requires the use of a backbone area, which connects all other areas in the AS and is always assigned an area ID of 0.0.0.0.

  Although an area ID has the same superficial form as an IP address, the area ID address space is its own distinct address space.

- **Stub area** — An OSPF area that does not accept or distribute external address advertisements. Instead, the area border router generates a default external route that is advertised into the stub area for destinations outside the autonomous system. Use the stub area designation to minimize topological data that is stored in the area's routers.
- **Range** — An address that covers a range of subnetwork addresses. A range address aggregates LSAs from all of its subnetwork addresses; this aggregation is also known as *route summarization*. For more information, see "Configuring Route Summarization in ABRs" later in this chapter.
- **Default route metric** — The network cost for an OSPF default route. If a default route metric is defined, the router advertises itself as the default router to the area. For more information, see "Default Route Metric" later in this chapter.

Types of Areas

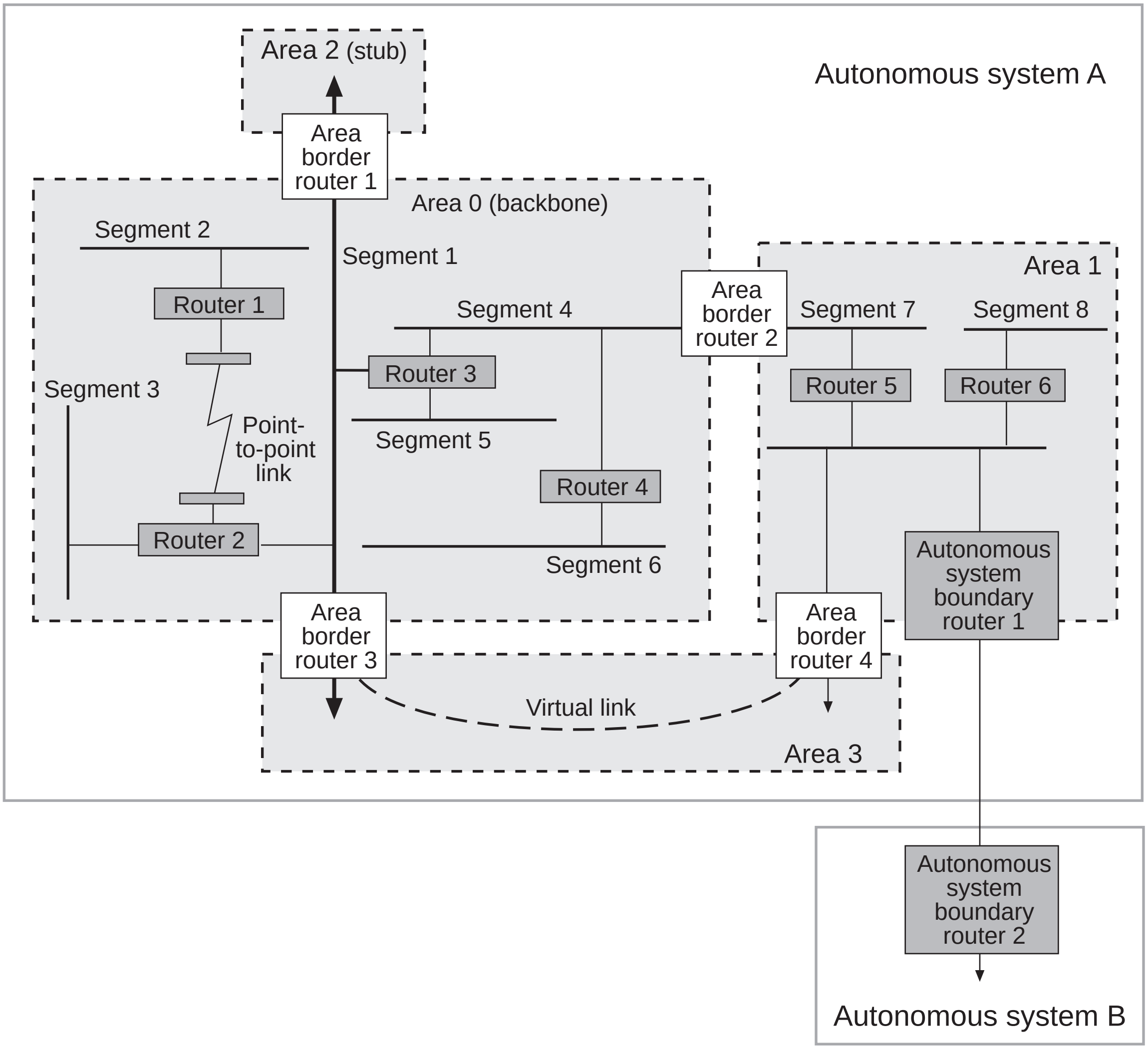
All routers within the same area maintain and use identical link state advertisement (LSA) databases. The network shown in Figure 54 later in this chapter contains four OSPF areas within autonomous system A. There are three types of OSPF areas:

- **Transit area** — An area through which network traffic can pass to reach other areas in the autonomous system. In Figure 54, the backbone area and areas 1 and 3 are transit areas.
- **Backbone area** — A contiguous area within an autonomous system that is divided into more than one area. The system defines the backbone area as 0.0.0.0. The backbone area distributes routing data between other areas in the autonomous system. By definition, the backbone area is also a transit area.
- **Stub area** — Generally, an area with only one entry or exit router. As a result, external routes are never flooded into stub areas. Instead, the area border router that is attached to the stub area advertises a single default external route into the area. This relationship conserves significant LSA database space that would otherwise be used to store external link state advertisements flooded into the area. In Figure 54, area 2 is a stub area that is reached only through area border router 1.

  It is possible to have a stub area with multiple area border routers and multiple exit points. However, all of the exit points and routers must contain the same external routing data so that the choice of an exit point does not need to be made for each external destination.

An area's network topology is visible only to systems inside that area; it is not visible to systems outside that area. Conversely, the systems within an area cannot see detailed network structures outside the area. Because of this restriction of topological information, you can control traffic flow between areas and reduce routing traffic to below the levels that occur when the entire autonomous system is a single routing domain.

Figure 54 Sample OSPF Routing Application



Area Border Routers

Each area (including the backbone area) includes all border routers that are connected to the area. In Figure 54, for example, you define:

- Area border routers 1, 2, and 3 as being in backbone area 0
- Area border routers 2 and 4 as being in area 1
- Area border router 1 as being in area 2
- Area border routers 3 and 4 as being in area 3

Routers must communicate with each other through interfaces that are defined as being in the same area. An area, however, may contain virtual links from area border routers to the backbone area. For example, in Figure 54, area border routers 3 and 4 terminate a virtual link between area 1 and the backbone area. For more information, see "Virtual Links" later in this chapter.

Routing Databases

All routers that are connected to an area maintain identical routing databases about the area. Routers that are connected to multiple areas maintain a separate routing database for each attached area. For example, in Figure 54:

- Routers 1, 2, 3, and 4 maintain identical routing databases about backbone area 0.
- Routers 5 and 6 maintain identical routing databases about area 1.
- Area border router 1 maintains separate routing databases about backbone area 0 and area 2.
- Area border router 2 maintains separate routing databases about backbone area 0 and area 1.
- Area border router 3 maintains separate routing databases about backbone area 0 and area 3.
- Area border router 4 maintains separate routing databases about areas 1 and 3. It also maintains a separate routing database about area 0 due to its virtual link to this backbone area through area border router 3.
- Autonomous system boundary routers 1 and 2 maintain separate routing databases about autonomous systems A and B.

Configuring Route Summarization in ABRs

The concept of route summarization is key in implementing a stable and scalable OSPF internetwork. *Route summarization* is the consolidating of advertised addresses by area border routers (ABRs). Instead of advertising routes to individual nodes within an area, you can configure an ABR to advertise a single summary route or “network range” that covers all the individual networks within its area that fall into the specified range.

Route summarization helps to control routing table sizes and prevents route flapping — that is, the constant changing of routes whenever an interface within an area comes online or goes offline. Using route summarization, routes within an area that change do not need to be changed in backbone ABRs and other area routers.

For optimal route summarization of an area, structure the area with a contiguous addressing scheme so that all routes within the area fall within a specified address range. This summary route or address range is defined by an IP address and mask combination. OSPF supports Variable Length Subnet Masks (VLSMs), so you can summarize a range of addresses on any bit boundary in a network or subnetwork address.

For example, an address range specified with an IP address of 142.194.0.0 with a mask of 255.255.0.0 describes a single route to a collection of destinations between 142.194.0.0 and 142.194.255.255.

Important Considerations

Consider the following guidelines when you design and configure areas:

- Define individual areas as contiguous, that is, any router that participates in OSPF exchanges must have a direct path to all other OSPF routers in the same area.
- Do not disconnect an area border router from the backbone area. This action may result in a loss of network topology information and routing capability.
- Define redundant links between area routers to help prevent area partitioning.
- If a portion of your internetwork consists of a particularly high number of nodes, consider creating an area specifically for that densely connected portion of your network.

- Whenever there is a change in network topology (such as when a link is lost or comes online), routers in all affected areas must converge on the new topology. If your internetwork consists of unstable links, you can partition the AS into smaller areas to minimize the number of areas that affected when the topology changes as a result of those unstable links.

Stub areas

- Because area border routers do not advertise external routes into stub areas, configuring an area as a stub area helps to minimize router table sizes, and therefore the memory requirements, of the routers within the area. Routers within a stub area need to store only a single default external route for destinations outside the autonomous system, as well as for intra-area and inter-area routes.
- If your network has no external routes, there is no advantage to configuring a stub area.
- Stub areas cannot contain autonomous system boundary routers (ASBRs)

Backbone area

A stable, fault-tolerant backbone is vital to your OSPF internetwork. It ensures communication between all areas within the AS. Consider the following guidelines when you design the backbone area:

- If you have only one area in your autonomous system, then you do not need to configure a backbone area (0.0.0.0).
- A backbone area must have the area ID of 0.0.0.0. Routers in the backbone area must be able to communicate with each other through interfaces that are configured in area 0.
- Connect all area border routers to the backbone area with either physical or virtual links.
- Backbones must be contiguous, meaning all area border routers (ABRs) that comprise the backbone must have a direct path to all other ABRs that are attached to the backbone.
- Configure ABRs with high redundancy so that no single link failure can cause a partition in the backbone.
- The more areas that the backbone interconnects, the greater the volume of traffic that must travel over the backbone. To increase stability, keep the size of the backbone reasonable.

- Because all routers connected to the backbone (ABRs) must recompute routes whenever the topology changes for any link in the AS, keeping the size of the backbone to a minimum is especially important in an autonomous system that may contain unstable links. At the very least, reducing the number of areas that connect a backbone directly reduces the likelihood of link-state change.
- Keep the maximum number of routers in the backbone area to about 50 or so, unless the link is of particularly high quality and the number of routes is minimal.
- Every ABR must connect to the backbone; this connection can be physical or virtual. If a router has an OSPF neighbor that is physically connected to the backbone, the router can use that neighbor to establish a virtual link to the backbone. Do not use too many virtual links to connect ABRs for the following reasons:
  - Stability of the virtual link depends on the stability of the underlying area that it spans.
  - This dependency on underlying areas can make troubleshooting difficult.
  - Virtual links cannot run across stub areas.
- Avoid placing hosts, such as workstations, servers, and other shared resources, within the backbone area.
- Having more than one ABR per area reduces the chance that the area will disconnect from the backbone.
- A single ABR can connect one or more areas to the backbone. To maximize stability, a single ABR should support no more than three areas because the router must run the link-state algorithm for each link-state change that occurs for every area to which the router connects.

Default Route Metric

An OSPF router always forwards an IP packet to the network that is the *best match* for the packet's destination; *best match* means the longest or most specific match. A router that fails to find a specific match for a packet's destination forwards the packet to the default router in the area.

To configure an OSPF router to advertise itself as the default router for an area, you define a default route metric. By default, the default route metric is not defined, which means that the router does not advertise itself as the area's default router.

When you remove the default route metric, the router no longer advertises itself as the default router.

OSPF Interfaces

You configure OSPF router interfaces by adding OSPF characteristics to existing IP VLAN interfaces. The OSPF interface has the following characteristics and statistics, which are discussed in the next sections:

- Mode
- Priority
- Area ID
- Cost
- Delay
- Hello Interval
- Retransmit Interval
- Dead Interval
- Password
- Statistics

Mode

The mode for an interface can be *off* or *active*. To run OSPF routing on the interface, set the mode to *active*.

Priority You assign the interface priority to an OSPF router to determine its status as a designated router. A router can function in one of three ways:

- **Designated router (DR)** — The router that has the highest priority value, unless a designated router already exists on the network segment.
- **Backup designated router (BDR)** — The router that has a lower priority than the DR; the BDR takes over DR functions if the DR fails.
- **Not a designated router** — Any router that is given a priority of 0 or that is not elected DR or BDR. Priority 0 routers can never be elected as a DR or a BDR.

Using Priority to Select a Designated Router

Each OSPF area on a broadcast or nonbroadcast multiaccess network that has at least two attached routers requires a designated router and a backup designated router. The designated router (DR) forms adjacencies to all other routers in the area. If for any reason the DR fails, the backup designated router takes over as the designated router.

To configure a router to be chosen as a designated router, you must understand how the designated router is elected:

- The routing interface that has the highest routing priority within an area is elected as the designated router using the Hello protocol.
- In case of a tie — two or more routers having the same highest routing priority — the router with the highest router ID is chosen as designated router.
- After a designated router is chosen, the same process is used to elect a backup-designated router, with the existing designated router excluded from the election.

Therefore, to designate a router to be elected as the designated router for an area, configure the router with a higher router interface priority than the other routers within the same area. If you want to prevent certain routers within an area from serving as a designated router or backup designated router, configure those routers with a router priority of “0,” because interfaces with a router priority of “0” are not eligible for designated router and backup designated router election.

When a router interface within an area first comes online, it determines if a designated router exists for the area. If one exists, the new router accepts the designated router regardless of its own router priority. Therefore, if you want to change the designated router for an area, configure the router that you want to serve as the new designated router to have a higher priority than other routers in the same area. The next time that the election process is initiated, the router that has the highest router priority is elected as the designated router for the area.

Area ID The area ID associates the router interface with an OSPF area. By default, all OSPF interfaces that you create on the system belong to the backbone area (0.0.0.0). If you want to change this association, specify a different area ID for any or all interfaces on the system.

Cost The interface cost parameter reflects the line speed of the port. Although the system calculates a default cost based on the module media type, you can set the cost manually to a different value. In most cases, you can accept the default value that the system sets.

Specifying Cost Metrics for Preferred Paths

In OSPF, the best path is the one that offers the least-cost metric. A cost is associated with each router output interface and each route as follows:

- Each output interface is assigned a default cost by the system based on the media bandwidth to which it is attached.
- Each route is assigned a metric that is equal to the sum of all metrics for all the links in the route.

You can configure area routers to use preferred paths by manually setting higher cost metrics for those paths that are not preferred.

For example, the fastest default media for OSPF is FDDI. All interfaces that are attached to a Fiber Distributed Data Interface (FDDI) media have a default cost metric of 1. All interfaces attached to faster media types, such as ATM or Gigabit Ethernet, are also assigned a default cost of 1. To ensure that the Gigabit Ethernet interface is the preferred route, leave the Gigabit Ethernet link with a metric cost of 1, and manually configure the slower links, such as FDDI, with a higher cost metric, for example, 2.

Delay The transmit delay is the estimated time (in seconds) that it takes for the system to transmit a link state update packet on the interface. The system increases the age of the link state advertisements (LSAs) that are contained in the update packets by the value that you specify for the delay.

This delay setting has more significance for interfaces that are connected to very low speed links because, on slower speed links, it is more probable that the router may send out back-to-back data packets more quickly than other routers and hosts can receive them. To avoid this situation, set the transmit delay to configure the router to wait a specified number or seconds between transmissions.

The delay value that you specify for an interface also increases the age of all LSAs that are transmitted over the interface by the same value. This setting may also affect how soon the LSA is flushed from an area router's database. Reasons that an LSA is flushed from a router's link state database include:

- **LSA is overwritten by a newer instance of the LSA.** For example, when a router receives similar LSAs (LSAs that have identical sequence and checksums), it then compares the ages of each LSA, and stores the LSA that has the least age value in the LSA database. This LSA is then used for routing table calculations.
- **LSA ages out.** When an LSA reaches the maximum age allowed by the system, the router first refloods the LSA onto the network. When it is no longer needed to ensure database synchronization (for example, when the LSA is no longer contained in neighbor LSAs), it is then flushed from the database.

Hello Interval The Hello interval (in seconds) determines how often the interface transmits Hello packets to other routers. The hello interval value must be identical among all routers that are attached to a common network. *Hello packets* notify other routers that the sending router is still active on the network. If a router does not send Hello packets for the period of time that is specified by the dead interval, that router is considered inactive by its neighbors, and all participating OSPF routers within the affected areas converge on the new topology. Therefore, the smaller the Hello interval, the faster that topological changes are discovered; as a result, however, more routing traffic occurs. The default value for the Hello interval is 10 seconds.

Retransmit Interval

When a router sends a link state advertisement to its neighbor, it keeps a copy of the LSA until the neighbor acknowledges receipt of the LSA with a link state acknowledgment packet. If the sending router does not receive a link state acknowledgment from its neighbor, it then retransmits the LSA. The retransmit interval (in seconds) determines how long the sending router waits for an acknowledgement before retransmitting the LSA to its neighbors.

To prevent needless retransmissions, the value that you specify must be greater than the roundtrip delay between any two routers on the attached network.

Dead Interval

The dead interval determines how long neighbor routers wait for a Hello packet before they determine that a neighbor is inactive. Each time that a router receives a Hello packet from a neighbor, the router resets the dead interval timer for that neighbor. The dead interval must be the same for all routers on the network. The default value for the dead interval is 4 times the default value for the Hello interval — 40 seconds.

Password

OSPF supports simple password authentication. You can set security passwords for OSPF interfaces so that only routers that know the password participate in OSPF exchanges. Therefore, configure routers in the same area that want to participate in the routing domain with the same password.

When you configure a password on a router interface, the interface inserts the specified password in the OSPF header of every packet that it transmits and receives only those OSPF packets that contain the same password.

Simple password authentication prevents routers from inadvertently joining the area and helps ensure that only trusted routers participate in the routing domain.

By default, OSPF interfaces on your system do not have associated passwords. When no password is assigned to an interface, OSPF exchanges are not authenticated so that, although a password may exist in an OSPF packet header, it is not examined when it is received.

Statistics

You can display interface statistics for diagnostic and network debugging purposes. Viewing the statistics for a particular interface can provide valuable information, such as whether the router is overburdened, and the number of Hello interval, dead interval, area ID, and password mismatches that the interface has seen on the network. For a complete listing of OSPF interface statistics, see the `ip ospf interface statistics` command in the *Command Reference Guide*.

Important Considerations

Consider the following guidelines when you configure router interfaces:

- To set the OSPF interface mode to active, enable IP routing.

Designated routers

- Because designated routers and backup designated routers have the most OSPF work to do within an area, select routers that are not already loaded with CPU-intensive activities to be the designated router and backup designated router.
- Because router priority is assigned on a per-interface basis, a single router with interfaces within several different areas can serve as designated router for those areas. But because a designated router has several CPU-intensive responsibilities, it is not a good idea to select the same router as designated router for many areas simultaneously.
- Routers that have an interface priority of 0 cannot serve as a designated router or backup designated router.
- On a broadcast network, if there is no designated router or backup designated router (such as when all routers have a priority of 0), routers do not form neighbor adjacencies, and routing information is not exchanged.

Area ID

- Set the area ID to the same value for all routers on the network segment. All routers in the same area *must* have the same area ID.
- The backbone area 0.0.0.0 is configured by default. The system associates all newly defined OSPF interfaces with the backbone area. You can change this association by changing the area ID for the selected interface.

Transmit delay

- The default value for the transmit delay is 1 second.
- Set the transmit delay to an integer value greater than 0.
- To set the transmit delay, take into account the transmission and propagation delays for the interface.
- Set the transmit delay according to the link speed; use a longer transmit delay for slower link speeds.
- The transmit delay is more effective on very low link speeds.

Hello interval

- The default value for the Hello interval is 10 seconds.
- The smaller the Hello interval, the faster that topological changes are detected, although more routing traffic ensues.
- Set the Hello interval to the same value for all routers on the same network segment.

Dead interval

- The default value for the dead interval is 40 seconds.
- Set the dead interval to 4 times the value specified for the hello timer.
- Set the dead interval to the same value for all routers on the same network segment.

Retransmit interval

- The default value for the retransmit interval is 5 seconds.
- Set the retransmit interval to greater than the expected round trip delay between any two routers on the attached network.
- Set the value that you specify for the retransmit interval conservatively, to avoid needless transmissions.
- Set the retransmit interval higher for serial lines and virtual links.

Password

- By default, an OSPF interface does not have an associated password.
- Use the same password for all routers on the same network segment.
- OSPF passwords are not encrypted. Therefore, a packet analyzer can be used to obtain the password by tapping the wire.
- If no password is defined for an interface, the interface does not verify the existence of a password on reception of a packet.

Link State Databases

OSPF routers use the information that is contained in the link state advertisements (LSAs) to build and maintain link state databases. Each link state database contains the link state advertisements from throughout the areas to which the router is attached. OSPF uses the following types of LSAs:

- Router Link State Advertisements
- Network Link State Advertisements
- Summary Link State Advertisements
- External Link State Advertisements

Router Link State Advertisements

All routers in an OSPF area originate router link state advertisements, also known as link state advertisements. Each link state advertisement describes the state and cost of the originating router's links (interfaces) to the area. Information contained in each link state advertisement includes:

- **LSID (Link State ID)** — The ID of the router that generated the LSA.
- **Router ID** — ID of the router that originated the LSA.
- **LS Seq (Link State Sequence)** — The sequence number of the advertisement. Used to detect old or duplicate link state advertisements.
- **LS age** — The time, in seconds, since the LSA was generated.
- **Flags** — Possible values:
  - **V** — Router is the endpoint of an active virtual link that is using the area as a transit area.
  - **ASBR** — Router is an autonomous system boundary router (ASBR).
  - **ABR** — Router is an area border router (ABR).
- **Link Type** — A description of the router link. Possible values:
  - **PTP** — Connection is point-to-point to another router.
  - **Transit** — Connection is to a transit network.
  - **Stub** — Connection to a stub network.
  - **Virtual link** — Connection is to a far-end router that is the endpoint of a virtual link.

- **Link ID** — Identifies the object to which this router link connects for each Link Type. Possible values:
  - **If Link Type is PTP**, then this is the neighboring router's router ID.
  - **If Link Type is Transit**, then this is the address of the designated router.
  - **If Link Type is Stub**, then this is the IP network or subnetwork number.
  - **If Link Type is Virtual Link**, then this is the neighboring router's router ID.
- **Link Data** — Provides additional link information. Possible values:
  - **If Link Type is PTP**, then this is the MIB II index value for an unnumbered point-to-point interface.
  - **If Link Type is Transit**, then this is the IP address of the advertising router's interface.
  - **If Link Type is Stub**, then this is the network's IP address mask.
  - **If Link Type is Virtual Link**, then this is the IP address mask of the neighboring router.
- **Metric** — Cost of using this outbound router link. With the exception of stub networks, this value must be other than 0.

Network Link State Advertisements

The designated router for each area originates a network link state advertisement for each transit network — a network that has more than one attached router. This advertisement describes all routers that are attached to the network, including the designated router itself. Each network link state advertisement (LSA) includes this information:

- **LSID (Link State ID)** — The ID of the router that generated the LSA.
- **Router ID** — ID of the router that originated the LSA.
- **LS Seq (Link State Sequence)** — The sequence number of the advertisement. Used to detect old or duplicate link state advertisements.
- **LS age** — The time, in seconds, since the LSA was generated.
- **Network Mask** — IP address mask for the network.
- **Attached Routers** — List of routers that are fully adjacent to the designated router (DR). The ID of the DR is also listed here.

Summary Link State Advertisements

Area border routers can generate two types of summary link state advertisements:

- Summary link state advertisements that report the cost to a single subnetwork number outside the area. These advertisements are identified as Type 3 in the link state advertisement header.
- Summary link state advertisements that report the cost to a single autonomous system boundary router (ASBR). These advertisements are identified as Type 4 in the link state advertisement header.

Each summary link state advertisement includes this information:

- **LSID (Link State ID)** — Possible values:
  - **For Type 3 summary link advertisements**, this is the IP network number.
  - **For Type 4 summary link advertisements**, this is the ASBR's router ID.
- **Router ID** — ID of the router that originated the LSA.
- **LS Seq (Link State Sequence)** — The sequence number of the advertisement. Used to detect old or duplicate link state advertisements.
- **LS age** — The time, in seconds, since the LSA was generated.
- **Network Mask** — For Type 3 summary link state advertisements, this is the destination network's IP address mask. For Type 4 summary link state advertisements, this parameter is set to 0.
- **Metric** — The cost of the specified route.

External Link State Advertisements

Each autonomous system boundary router generates an external link state advertisement for each network destination (known to the router) outside the AS. AS boundary routers use these external link state advertisements to describe routes to destinations outside the AS. For these advertisements, the Link State ID field in the advertisement header specifies an IP address.

In addition, OSPF also considers the following routes to be external routes. They are advertised using external link state advertisements:

- The default route
- Static routes
- Routes derived from other routing protocols, such as RIP
- Directly connected networks that are not running OSPF

All external routes are assigned a cost metric. External route cost is calculated based on one of these cost metric types:

- **Type 1** — Router adds the internal cost metric to the external route metric. For example, if an ABR is advertising Type 1 external route metrics, the cost of the route from any router within the AS is equal to the cost associated with reaching the advertising ABR, plus the cost of the external route.
- **Type 2** — Routers do not add the internal route metric to the external route metric. Therefore, the router that advertises the smallest external metric is chosen, regardless of the internal distance to the AS boundary router. For example, if an ABR is advertising Type 2 external route metrics, the cost of the route from any router within the AS is equal to the cost of the external route alone. The cost of reaching the advertising ABR is not considered in determining the cost of the external route. The internal cost is only used as a *tie-breaker* when several equal-cost Type 2 routes exist.

When both Type 1 and Type 2 metrics are present in an AS, Type 1 external metrics take precedence.

Each external link state advertisement includes this information:

- **LSID (Link State ID)** — An IP network address:
  - **For Type 3 summary link advertisements**, this is the IP network number.
  - **For Type 4 summary link advertisements**, this is the ASBR's router ID.
- **Router ID** — ID of the router that originated the LSA.
- **LS Seq (Link State Sequence)** — The sequence number of the advertisement. Used to detect old or duplicate link state advertisements.
- **LS age** — The time, in seconds, since the LSA was generated.
- **Network Mask** — The IP address mask for the advertised destination.
- **Fwd address (Forwarding Address)** — If the AS boundary router is advertising a destination that can be more optimally reached by a different router on the same LAN, then the advertising boundary router specifies that router's address in the forwarding address field. Otherwise, it leaves the field as 0.
- **Metric** — The cost to reach the advertised destination.
- **Type** — Possible values:
  - **Type 1** — Normal link state metric.
  - **Type 2** — The metric is larger than any local link state path. See the discussion of Type 1 and Type 2 external metrics earlier in this section.
- **Route Tag (External)** — Not used by OSPF. These 32-bits may be used to communicate other information between boundary routers.

Important Considerations

When you view the link state database, consider the following:

- An asterisk (*) after the router ID in a display indicates that the LSA originated locally.
- All routers within an area must maintain identical link state databases.
- Use the contents of the link state database for network configuration and troubleshooting purposes.

Neighbors

Neighbor routers are those that are physically attached to the same network segment. The OSPF Hello protocol establishes adjacencies among neighboring routers to facilitate the exchange of routing information. An *adjacency* describes the relationship between two routers that exchange network topology information. A router attached to multiple network segments may have different sets of neighbors on each segment.

For example, Figure 54 earlier in this chapter includes several sets of OSPF neighbor routers. In backbone area 0:

- Routers 2 and 3 and area border routers 1 and 3 are neighbors on segment 1 (the backbone network).
- Routers 1 and 2 are neighbors on a point-to-point link.
- Routers 3 and 4 and area border router 2 are neighbors on segment 4.
- No routers are neighbors on segments 2, 3, 5, and 6.

In area 1:

- Router 5 and area border router 2 are neighbors on segment 7.
- Routers 5 and 6, area border router 4, and autonomous system boundary router 1 are neighbors on segment 9.
- No routers are neighbors on segment 8.

In area 3, area border routers 3 and 4 are neighbors on a virtual link between the backbone area 0 and area 1.

Neighbor Information

Your system can display a list of all neighbors for all OSPF interfaces defined on the system. The list includes the following information:

- **Index** — The Index number that corresponds to the OSPF router interface for which neighbors have been discovered.
- **Neighbor Address** — The IP address of the neighboring router.
- **Router ID** — The router ID of the neighboring router.
- **State** — The state of the adjacency. You can also think of this as the state of the conversation that is held with the neighboring router. Possible neighbor state values:
  - **Down** — The initial state of a neighbor conversation. It indicates that no recent information has been received from this neighbor.
  - **Attempt** — Only used on nonbroadcast networks. This value indicates that no recent information has been received from this neighbor, but the router tries to contact the neighbor by sending Hello packets.
  - **Init** — A Hello packet has recently been seen by a neighbor, but two-way communication has not been established.
  - **Two-way** — Bidirectional communication has been established. Two-way is the most advanced state of a neighbor relationship before beginning to establish an adjacency. In fact, the designated router and backup designated router are selected from the set of neighbors that are in a state of two-way communication or greater.
  - **Exstart** — The initial step in creating an adjacency between two routers. Adjacencies involve a master/slave relationship between two routers, which is when that relationship is determined. The master sends the first information describing its link state database in the form of database description packets. The slave can only respond to the database description packets.
  - **Exchange** — The router is describing its link state database by sending database description packets to the neighbor. All adjacencies in the exchange state are used by the flooding procedure. Adjacencies in this state are capable of transmitting and receiving all types of OSPF protocol packets.

  - **Loading** — The router is sending requests for link state advertisements (LSAs) that were discovered in the exchange state but not yet received.
  - **Full** — The neighbor is now fully adjacent. This adjacency is now advertised in router LSAs and network LSAs.
- **Priority** — The priority of this neighbor in terms of designated router election. A value of 0 indicates that the neighbor is not eligible to become a designated router.
- **RxQ (Retransmit Queue)** — The number of LSAs in the local retransmit queue to the neighbor. These LSAs have been flooded but not acknowledged on this adjacency. The LSAs in the queue are flooded until they are acknowledged by the neighbor or until the adjacency is destroyed.
- **SumQ (Summary Queue)** — The number of LSAs that make up the area link state database at the moment that the neighbor goes into the database exchange state. These LSAs are sent to the neighbor in database description packets.
- **ReqQ (Request Queue)** — The number of LSAs that are required from the neighbor in order to synchronize the neighboring routers' link state databases. The router requests these LSAs by sending link state request packets to the neighbor. The neighbor then responds with link state update packets containing the requested LSAs. As the appropriate LSAs are received from the neighbor, they are removed from the request queue.
- **Flags** — The type of neighbor. Possible values:
  - **D** — The neighbor was dynamically discovered.
  - **S** — The neighbor was statically defined.
  - **BDR** — The neighbor is the backup designated router for the area.
  - **DR** — The neighbor is the designated router for the area.
- **Examples**
  - `S+BDR` indicates that the neighbor was statically defined and serves as the backup designated router for the area.
  - `D+DR` indicates that the neighbor was dynamically discovered and serves as the designated router for the area.

Static Neighbors

On broadcast networks such as Ethernet, the OSPF Hello protocol uses the broadcast capability to dynamically discover neighbors. On nonbroadcast networks, such as X.25 Public Data Network, however, you may need to assist in neighbor discovery by statically defining neighbors on each interface. OSPF then uses the Hello protocol to maintain the neighbors that you statically define.

When you statically define a neighbor on the system, you specify both the router interface to which you want to add the neighbor and the IP address of the neighboring router that you want to associate with the specified interface. The Hello protocol then dynamically retrieves the additional neighbor information, as described in "Neighbor Information" in the previous section.

Important Considerations

Consider the following guidelines when you configure neighbors:

- Routers use OSPF hello packets to learn neighbor addresses dynamically on broadcast networks.
- Define static neighbors only on nonbroadcast interfaces, because neighbors are not learned dynamically on nonbroadcast networks.
- Hello packets are the only OSPF packet type that is processed by all routers within an area. All other packet types are sent and received only on adjacencies.
- Neighbor adjacencies cannot be formed if two routers have different Hello intervals, Dead intervals, or passwords.

Router IDs

Each router that is configured for OSPF has an OSPF router ID. The OSPF router ID uniquely identifies the router to other routers within an autonomous system.

The router ID determines the designated router in a broadcast network if the priority values of the routers involved in the designated router election are equal. In the event of a *priority* tie, the router with the highest router ID is elected designated router for the area.

Three types of router identifiers, in the form of an IP address, are available:

- **Default** — A unique ID that the system generates and uses as the default router ID.
- **Interface** — The index of an IP interface on the router.
- **Address** — An ID that you define in the form of an IP address.

OSPF routing must be inactive before you can add or modify an OSPF router ID. To deactivate OSPF routing, set the OSPF mode to `disabled`. See the *Command Reference Guide* for details. After you add the router ID, you can set the OSPF mode to `enabled` on the interface.

Important Considerations

Consider the following guidelines when you configure OSPF router IDs:

- For OSPF to operate correctly, the router ID must be unique for every router.
- Choose the default setting to ensure unique router IDs.
- You cannot set the router ID to either 0.0.0.0 or 255.255.255.255.

OSPF Memory Partition

There are three choices for OSPF memory allocation:

- Have the system intelligently determine the maximum OSPF memory partition size (partition size = `1`). This is the default.
- Have OSPF be part of system memory, growing as needed and without limit (partition size = `0`).
- Configure the maximum OSPF memory partition size manually (partition size = `4096` - `<maximum available memory>`).

You use the `ip ospf partition modify` option to control memory allocation, as described in the *Command Reference Guide*.

Default Memory Allocation

You typically do not have to modify the OSPF memory allocation. By default, the system manages memory by partitioning the total memory available for applications among the various protocols. This functionality ensures that the router has enough memory for to perform all of its functions and enable most features. Under this option, OSPF always has a partition of memory available for its use.

Under the default OSPF memory allocation scheme, two values have meaning:

- Current partition maximum size
- Allocated memory size

Current Partition Maximum Size

The *current partition maximum size* is the maximum amount of memory that OSPF can allocate. It is calculated at system startup as a function of the maximum routing table size and available memory by the following formula:

(((externalLSAsize * maxRoutingTableSize + 100000) / 100000) + 1) * 100000

Because most of the routes are going to be external to OSPF, the formula bases the OSPF memory partition maximum size on the amount of memory that is required to store an external link state advertisement (LSA), 80 bytes, times an estimate of the maximum number of routing table entries that the system can hold (maxRoutingTableSize). It then rounds to the nearest 100000 bytes and adds an additional 100000 bytes as a buffer.

The estimate (maxRoutingTableSize) of the maximum number of routing table entries the system can hold for a given memory size is a hardcoded value. On extended memory systems this value is 51200. On systems without extended memory this value is only 1024.

Applying the formula to extended memory systems yields a default OSPF current partition maximum size of 4,200,000. (Due to memory overhead, the actual number of routing table entries possible is somewhat different than the 51200 maximum.)

Even though currently unallocated, this memory is not available to other protocols.

Allocated Memory Size

The *allocated memory size* is the size of the memory that is currently allocated to OSPF. The minimum size this allocated memory partition can default to is 100000.

The system allocates more memory as required in 100000-byte chunks until the current partition maximum size is reached.

Running Out of Memory — Soft Restarts

An attempt to allocate memory past the OSPF current partition maximum size generates a soft restart condition that momentarily causes the router to go down. This may occur, for example, because:

- The routing table grew suddenly because it received a large number of external link state advertisements (LSAs), such as RIP routes learned from an ASBR, that had to be added to the internal database.
- The router is an area border router (ABR) for multiple large subareas and thus has a much larger than usual routing table.

The `ip ospf statistics` *option displays the number of soft restarts.*

After the soft restart, the system frees all of its OSPF memory, disables its interfaces, reenables them, and reconstructs the router tables from scratch. This process attempts to free and defragment enough unused memory so that OSPF has sufficient memory to continue. If the soft restart does not free enough memory, the soft restart condition repeats — and the router continues to thrash for memory.

If the softRestarts statistic shows that the default memory allocation scheme is too small for your router, then you must use one of the other two memory allocation options described next.

Manual Memory Allocation

You can manually control the OSPF current partition maximum size. You can enter any value between 4096 and the maximum memory available on your system, as shown in the `ip ospf partition modify` command prompt.

You can also use manual memory allocation control to *lower* the OSPF current maximum partition size to be less than the 4,200,000 default minimum on extended memory systems. As noted previously, memory reserved under the OSPF current maximum partition size is not available to other protocols even if it is not allocated. If you must carefully apportion memory among competing protocols, then you might want to decrease the memory available to OSPF. A router located in a stub area has no external link state advertisements (LSAs), for example, and might require less memory.

System Memory Allocation

You can also have OSPF use the *system* memory partition. There will be no specific OSPF memory partition and no current maximum partition size. OSPF will grow as it finds necessary, possibly encroaching upon the space available to other protocols.

Stub Default Metrics

Generally, a stub area is a network that is connected to an OSPF routing domain by a single area border router (ABR). External link state advertisements are not advertised into stub areas. Instead, the ABR injects a Type 3 summary link state advertisement that contains a single external default route into the stub area. The routers within the stub area use this single external route to reach all destinations outside the stub area. This arrangement saves routing table space and system resources because stub area routers do not have to learn a multitude of external routes for the greater network; they need only store a single external route.

The stub default metric determines whether an ABR generates the default route into the stub area to which it is connected, and the cost associated with that route.

For example, in Figure 54 earlier in this chapter, you would configure area border router 1 to generate a default route into stub area 2. If you define a stub default metric of 4, area border router 1 will generate a default route with an associated cost of 4 into stub area 2.

If you remove the stub default metric, the ABR does not advertise a default route into the stub area.

A stub area can have multiple ABRs and multiple exit points. However, all of the exit points and routers must contain the same external routing data so that the choice of an exit point does not need to be made for each external destination.

Important Considerations

Consider the following guidelines when you define stub default metrics:

- By default, area border routers advertise a stub default metric of 1.
- Stub default metrics are relevant only for area border routers (ABRs) that are attached to stub networks.
- If your network does not have external routes, you do not need to configure the stub default metric; and you do not need a stub area.
- If you remove the stub default metric, the ABR does not advertise a default route into the stub area.

Virtual Links

The backbone area (0.0.0.0) must link to all areas. If any areas are disconnected from the backbone, some areas of the autonomous system (AS) become unreachable. In the rare case that it is impossible to physically connect an area to the backbone, you can use a virtual link. The virtual link provides a logical path to the backbone for the disconnected area.

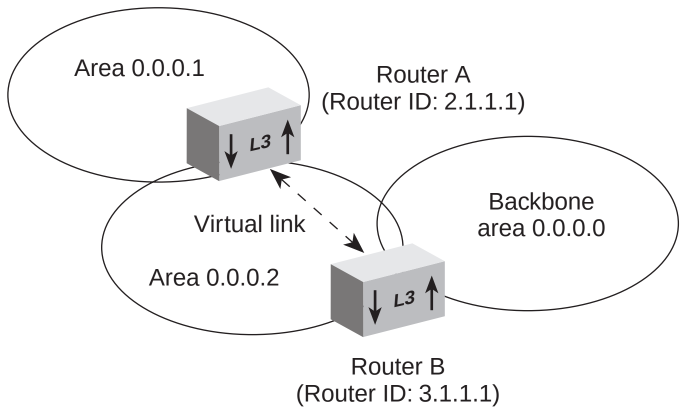
Virtual links are used to ensure that the OSPF backbone is contiguous. You can use virtual links to:

- Introduce new areas that do not have physical access to the backbone.
- Add redundancy to the backbone to help prevent partitioning.
- Patch the backbone when discontinuity occurs.
- Connect area backbones. For example, you can merge two existing OSPF networks into one network sharing a common backbone.

A virtual link must be established between two ABRs that share a common nonbackbone area, with one of those ABRs directly connected to the backbone. The nonbackbone area through which the virtual link runs is called a *transit area*.

The endpoints of a virtual link must be area border routers. You must configure the virtual link on both routers. Each router's virtual link definition includes the other router's router ID and the transit area through which the routers connect. Figure 55 illustrates a virtual link between two area border routers.

Figure 55 Virtual Link



In Figure 55, area 0.0.0.1 cannot be physically connected to the backbone area. Instead, connectivity to the backbone is achieved using a virtual link, configured between router A and router B. Area 0.0.0.2 is the transit area, and router B is the entry point into backbone area 0.0.0.0. The virtual link in Figure 55 provides area 0.0.0.1 and router A with a logical connection to the backbone. Here is the virtual link configuration for both routers shown in Figure 55:

- Router A:
  - Transit area: 0.0.0.2
  - Target router: 3.1.1.1
- Router B:
  - Transit area: 0.0.0.2
  - Target router: 2.1.1.1

Important Considerations

Consider the following guidelines when you configure virtual links:

- You must configure a virtual link for any area border router that has an interface connected to a location outside the backbone area.
- You can define up to 32 virtual links per router.
- You cannot configure a virtual link through a stub area.
- Use virtual links sparingly for the following reasons:
  - Stability of the virtual link depends on the stability of the underlying area that it spans.
  - This dependency on underlying areas can make troubleshooting difficult.

OSPF Routing Policies

Routing policies are rules that define criteria to control the flow of routes to and from the routing table. Your system supports two types of OSPF routing policies: *import* policies that dictate which routes are added to the routing table and *export* polices that dictate which routes are advertised to other routers. You can use routing policies to:

- **Increase security** — For security reasons, you may not want the router to advertise certain routes. For example, Organization A may have defined one of its ASBRs with a direct connection to Organization B that they use for direct communication. For security or performance reasons, A may not want to give other groups access to that connection. To prevent this direct connection from being known to other organizations, A can define an export policy that prohibits its ASBR from advertising the direct connection that it uses to communicate with B.
- **Conserve routing table space** — The selective nature of routing policies can minimize routing table sizes and increase network stability. For example, you may want to limit the number of hosts and gateways from which routing information is accepted, in which case you can define an import policy to selectively rule out, or reject, unnecessary routing table entries.

- **Isolate suspect networks** — Misconfigured hosts can sometimes send inappropriate routing information, which can compromise network integrity. In such a case, you can define an import policy on an ASBR that rejects all routes from the suspect network.
- **Adjust route cost** — Both import and export policies let you change the cost that is associated with routes without physically changing the cost of an interface. For example, router A may advertise a route with one cost, but router B may use an import policy to write the same route to its routing table with a different, or adjusted, cost. Similarly, router A may have a route in its routing table with one cost but choose to advertise the route to other routers with a different cost.

Important Considerations

Consider the following guidelines when you work with OSPF routing policies:

- You can only apply OSPF policies against external routes. External routes refer to routes that are advertised over the network using external link state advertisements (LSAs). These routes include:
  - **Directly connected non-OSPF interfaces** — Physical interfaces on the router itself that are not running OSPF and that are directly connected to the network.
  - **RIP routes** — Routes to destinations outside the autonomous system learned via the Routing Information Protocol (RIP) and imported by autonomous system boundary routers.
  - **Static routes** — IP routes statically defined by the user.
- You cannot apply export policies against directly connected OSPF interfaces, because all routers in the area must maintain identical link state databases.

- With the ability to wildcard policy parameters (such as 0.0.0.0 to indicate all routers or all routes), occasions may arise when several policies match a route. In such cases, routers use the following procedure to determine which policy to apply to the route:
  - If multiple policies apply to the route, the router uses the policy that has the highest administrative weight.
  - If multiple matches still exist, the router uses the policy that matches the specified source (excluding wildcards).
  - If multiple matches still exist, the router compares route address bits and uses the policy that best matches the route address (excluding wildcards).
  - If multiple matches still exist, the router uses the policy that matches the origin protocol.
  - If multiple matches still exist, then the router uses the policy that has the lowest index number.
- You can set up an IP RIP or OSPF import or export policy to accept or advertise the default route, as long as the default route exists in the routing table. When you define a policy, you are always prompted for the route subnet mask after the route address, even though you specify the wildcard route address of 0.0.0.0.

  Specify a route subnet mask as follows:
  - If you want the wildcard subnet mask for all routes, enter 0.0.0.0 for the subnet mask. This is the default subnet mask.
  - If you want the default route (not all routes), enter 255.255.255.255.
- For more information about IP routing policies, see Chapter 16.

Implementing Import Policies

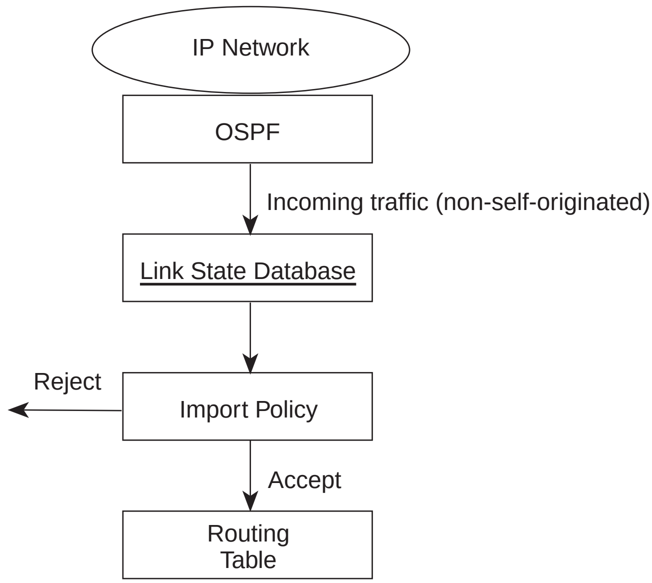
Import policies control which non-self-originated routes (RIP external routes) are accepted and stored in the routing table. *Non-self-originated* means that the router itself did not originate the route; it learned the route from an external link state advertisement. You can also adjust the cost of each route that is accepted into the routing table. Using RIP, you can define which *external routes* (for example, RIP) a router advertises and imports into the OSPF link state database. They are self-originated but must be external to OSPF. Do not reset these modules for several minutes after making configuration changes.

Because all routers within the same OSPF area must maintain similar databases, all routers must receive all link state advertisements that are sent over the network, and store those advertisements in their link state databases. By defining an import policy, however, you can control what routes from a router's external link state database are migrated to its routing table.

When you define an import policy, you specify a set of criteria. When a the router receives an external link state advertisement, it consults its routing policies to determine if the route specified in that advertisement matches any of the defined policies criteria, and, if so, applies to the route the actions that are defined by the policy.

Figure 56 illustrates the import policy process.

Figure 56 Import Policy Process



Information that you define for an import policy includes:

- The route or routes to which you want the policy to apply, specified by a network address and subnet mask.
- The action that you want the router to take — accept or reject. `Accept` configures the router to add the route to its routing table. `Reject` prevents the router from adding the route to its routing table.

- For routes that are accepted into the routing table as defined by the policy, you can define a new cost metric value for the route, or you can adjust the existing cost metric using one of these operators:
  - + adds the specified number to the existing cost metric
  - - subtracts the specified number from the existing cost metric
  - * multiplies the specified number by the existing cost metric
  - / divides the existing cost metric by the specified number
  - % modulo divides the existing cost metric by the specified number and returns the remainder

  The routes are then accepted into the routing table with the cost metric that has been defined by the import policy.
- In case multiple policies match the same route, you can also assign an administrative weight to define an order of precedence.

Import Policies at a Glance

Table 80 lists the possible import policy configurations.

Table 80 OSPF Import Policies

Route Address	Route Subnet Mask	Policy Action	Metric Adjustment	Description
0.0.0.0	N/A	Accept	C	Adds all external routes to routing table and assigns a cost of C to each
A	Subnet mask for route A	Accept	C	Adds Type 1 and Type 2 Route A to the routing table with an associated cost of C
0.0.0.0	N/A	Reject	N/A	Does not add any external routes to the routing table
A	Subnet mask for route A	Reject	N/A	Does not add Type 1 or Type 2 Route A to the routing table

Import Example 1: Accept Route

The policy defined in Table 81 imports route 243.140.28.0 into the routing table and assigns a cost of 10 to the route.

Table 81 Import Policy Example

Policy Field	Definition
Policy type	import
Route address	243.140.28.0
Route subnet mask	255.255.255.0
Policy action	accept
Metric adjustment	10
Administrative weight	16

Import Example 2: Reject Route

The policy defined in Table 82 prohibits the router from adding route 243.140.28.0 to its routing table.

Table 82 Export Policy Example

Policy Field	Definition
Policy type	import
Route address	243.140.28.0
Route subnet mask	255.255.255.0
Policy action	reject
Administrative weight	15

Implementing Export Policies

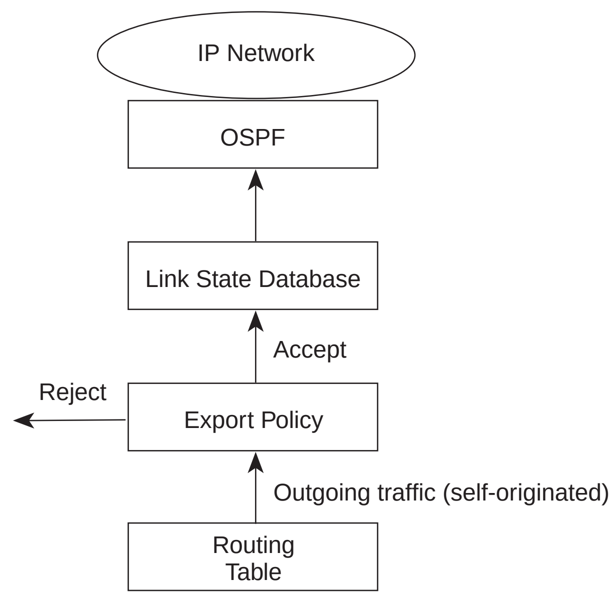
Using export policies, you can define which self-originated external routes a router advertises. *Self-originated* refers to routes that are originated by the router itself. You can also adjust the cost and external metric type of each route that you allow the router to advertise.

See the discussion about Type 1 and Type 2 metrics in "External Link State Advertisements" *earlier in this chapter for more information about external metric types.*

When you define an export policy, you can configure the router to accept or reject routes. An *accept* export policy configures the router to place the specified route in external link state advertisements for propagation over the network. The routes are advertised with the cost and the external metric type defined by the policy. A *reject* export policy prevents the router from placing the specified route in external link state advertisements, thereby prohibiting propagation of the route over the network.

Figure 57 illustrates the export policy process.

Figure 57 Export Policy Process



You define these criteria as part of an export policy:

- The method by which the route was learned by the router. Possible origins include directly connected interfaces and static routes, as well as RIP routes imported by autonomous system boundary routers.
- When you define an export policy against a directly connected interface, you can specify one or all of the physical router interfaces that are directly connected to the network against which you want the export policy to be applied. For example, if you define an export policy that rejects a direct interface, the router does not advertise the specified interface over the network.

- When you specify RIP or static as the origin protocol, you can specify the source address of the router that originated the RIP or static route. For example, you can define an export policy to reject (that is, not advertise) all statically defined routes, in which case you specify the local router's ID as the source address.
- The route or routes to which you want the policy to apply, specified by a network address and subnet mask.
- The action that you want the router to take:
  - **Accept** — The specified route is placed in external link state advertisements and propagated over the network.
  - **Reject** — The specified route is not placed in external link state advertisements and as a result is not propagated over the network.
- For export policies that define routes to be advertised in external LSAs, you can define a new cost metric value for the route, or you can adjust the existing cost metric using one of these operators:
  - + adds the specified number to the existing cost metric
  - - subtracts the specified number from the existing cost metric
  - * multiplies the specified number by the existing cost metric
  - / divides the existing cost metric by the specified number
  - % modulo divides the existing cost metric by the specified number and returns the remainder

  The routes are then advertised with the cost metric as defined by the export policy.
- You can choose to advertise the route as a Type 1 or a Type 2 external cost metric.
- In case multiple policies match the same route, you can also assign an administrative weight to define an order of precedence.

Export Policies for RIP and Static Routes

Table 83 shows the export policies that can be applied to RIP and statically defined routes.

Table 83 OSPF Export Policies for RIP and Static Routes

Origin Protocol	Source Router	Route	Policy Action	Metric Adjustment	External Metric Type	Description
RIP or Static	A	B	Accept	C	Type 1, Type 2	RIP or Static Route B originating from Router A is advertised as the specified metric type with a cost of C.
RIP or Static	A	0.0.0.0	Accept	C	Type 1, Type 2	All RIP or Static routes originating from Router A are advertised as the specified metric type with a cost of C.
RIP or Static	0.0.0.0	B	Accept	C	Type 1, Type 2	RIP or Static Route B originating from any router is advertised as the specified metric type with a cost of C.
RIP or Static	0.0.0.0	0.0.0.0	Accept	C	Type 1, Type 2	RIP or Static routes originating from any router are advertised as the specified metric type with a cost of C.
RIP or Static	A	B	Reject	N/A	N/A	RIP or Static Route B originating from Router A is not advertised.
RIP or Static	A	0.0.0.0	Reject	N/A	N/A	All RIP or Static routes originating from Router A are not advertised.
RIP or Static	0.0.0.0	B	Reject	N/A	N/A	RIP or Static Route B originating from any router is not advertised.
RIP or Static	0.0.0.0	0.0.0.0	Reject	N/A	N/A	RIP or Static routes originating from any router are not advertised.

Export Policies for Direct Interfaces

Table 84 shows the possible export policies that can be applied to directly connected router interfaces.

Table 84 OSPF Export Policies for Directly Connected Interfaces

Origin Protocol	Interface	Policy Action	Metric Adjustment	External Metric Type	Description
Direct	Specific non-OSPF interface or All non-OSPF interfaces	Accept	C	Type 1, Type 2	The specified interfaces are advertised as a Type 1 or Type 2 metric with a cost of C.
Direct	Specific non-OSPF interface or All non-OSPF interfaces	Reject	N/A	N/A	Do not advertise the specified interfaces.

Export Example 1: Prohibit Advertisement of non-OSPF Interfaces

The policy defined in Table 85 prohibits an autonomous system boundary router from advertising any directly connected non-OSPF interfaces.

Table 85 Export Policy to Reject Direct Interfaces

Policy Field	Definition
Policy type	export
Origin Protocol	dir
IP interfaces	all
Policy action	reject
Administrative weight	1

The router prohibits the address of any of its directly connected RIP interfaces from being placed in external link advertisements. As a result, the interfaces are not advertised over the network.

Because all OSPF routers must maintain similar link state databases and shortest path trees, you cannot define an export policy to restrict the advertisement of directly connected OSPF interfaces.

Export Example 2: Prohibit Advertisement of Static Address

The policy defined in Table 86 prohibits a router from advertising any static route originating from router 131.141.127.7.

Table 86 Export Policy to Reject Static Routes

Policy Field	Definition
Policy type	export
Origin protocol	sta
Source address	131.141.127.7
Route address	0.0.0.0
Policy action	reject
Administrative weight	1

Although the router can learn all static routes that originate from router 131.141.127.7, this policy prohibits any of those routes from being placed in external link advertisements.

Export Example 3: Prohibit Advertisement of RIP Routes

The policy defined in Table 87 prohibits an autonomous system boundary router from advertising imported RIP route 138.140.9.0 originates from router 131.141.126.9.

Table 87 Export Policy to Reject RIP Routes

Policy Field	Definition
Policy type	export
Origin protocol	rip
Source address	131.141.126.9
Route address	138.140.9.0
Route subnet mask	255.255.255.0
Policy action	reject
Administrative weight	1

Although the router can add the 138.140.9.0 route to its routing table, this policy prohibits the boundary router from migrating the route from its routing table to its link state database. As a result, the route is not propagated over the network.

Export Example 4: Advertisement of Direct Interfaces

The policy defined in Table 88 configures a router to advertise direct interface 8 as a Type 2 external metric with a cost increase of 2.

Table 88 Export Policy to Accept a Direct Interface

Policy Field	Definition
Policy type	export
Origin protocol	dir
IP interfaces	8
Policy action	accept
Metric adjustment	+2
ASE Type	Type 2
Administrative weight	1

Suppose a routing table entry exists for interface 8 that identifies the route as a Type 1 external metric with an associated cost of 10. This policy configures the router to export direct interface 8 from its routing table and write the routing interface information to its link state database as a Type 2 external metric, with an associated cost of plus 2. As a result the router advertises the interface over the network as a Type 2 external metric with an associated cost of 12, overriding the external metric type and cost that are defined for the interface in the system's routing table.

Export Example 5: Advertisement of Static Routes

The policy defined in Table 89 configures a router to advertise all static routes as Type 1 external metrics with a cost of 1.

Table 89 Export Policy to Accept Static Routes

Policy Field	Definition
Policy type	export
Origin protocol	sta
Source address	0.0.0.0
Route address	0.0.0.0
Policy action	accept
Metric adjustment	1
ASE Type	Type 1
Administrative weight	1

Export Example 6: Advertisement of RIP Routes

The policy defined in Table 90 configures an autonomous system boundary router to advertise all routes that are imported from a RIP network as Type 2 external metrics with associated costs of 10.

Table 90 Export Policy to Accept RIP Routes

Policy Field	Definition
Policy type	export
Origin protocol	rip
Source address	0.0.0.0
Route address	0.0.0.0
Policy action	accept
Metric adjustment	10
ASE Type	Type 2
Administrative weight	1

OSPF Statistics

From the Administration Console and the Web Management interface, you can display general statistics for specific OSPF interfaces. These statistics provide valuable information useful in troubleshooting network and system issues. For example, the number of SPF computations directly corresponds to the number of topological changes that the interface had to converge on. An excessive number of soft restarts may be an indication that the router is overburdened because of resource limitations. OSPF statistics:

- **SPF computations** — Number of shortest-path-first computations made. Each time that a router comes online, or each time there is a change in topology, the router must perform SPF computations.
- **Memory failures** — Number of nonfatal memory allocation failures.
- **LSAs transmitted** — Number of link state advertisements transmitted.
- **LSAs received** — Number of link state advertisements received.
- **Route update errors** — Number of nonfatal routing table update failures.
- **Receive errors** — Number of general receive errors.
- **External LSA changes** — Number of external LSA changes made to database.
- **Soft restarts** — Number of OSPF router soft restarts due to insufficient memory resources (implies a fatal memory allocation failure). To fix this problem, change the OSPF memory partition with the ip ospf partition modify option, add memory, or reconfigure the network topology to generate smaller OSPF databases.

Standards, Protocols, and Related Reading

OSPF as implemented on this system is described in the following Internet Engineering Task Force (IETF) Request for Comment (RFC) documents:

- RFC 1583, Moy, J., *OSPF Version 2*, March 1994.
- RFC 1850, Baker, F., and Coltrun, R., *OSPF Version 2 Management Information Base*, November 1995.

Other useful reading includes:

- Moy, John, *OSPF: Anatomy of an Internet Routing Protocol*, Reading, MA., Addison-Wesley/Longman, ISBN 0201634724, 1997.
- RFC 1245, Moy, J., *OSPF Protocol Analysis*, July 1991.
- RFC 1586, DeSouza, O., and Rodriguez, M., *Guidelines for Running OSPF Over Frame Relay Networks*, March 1994.

20 IPX ROUTING

This chapter provides guidelines and other key information about how to implement Internet Packet Exchange (IPX) protocol routing on Multilayer Switching Modules. The chapter covers these topics:

- IPX Routing Overview
- Key Concepts
- Key Guidelines for Implementation
- IPX Interfaces
- IPX Routes
- IPX Servers
- IPX Forwarding
- IPX RIP Mode
- IPX SAP Mode
- IPX Statistics
- Standards, Protocols, and Related Reading

After you log in to the system and connect to a slot that houses a Multilayer Switching Module, you can manage IPX routing features from the `ipx` *menu of the Administration Console. See the* Switch 4007 Command Reference Guide*.*

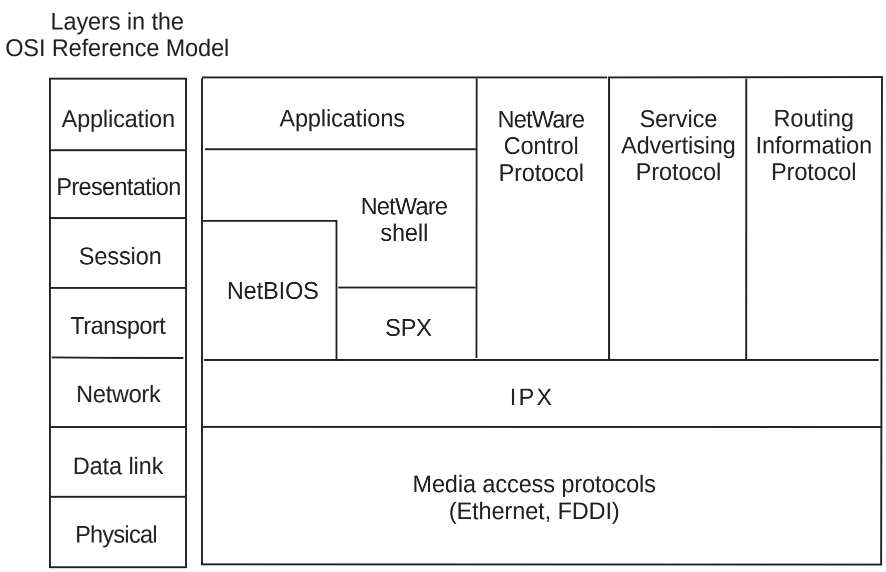
The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

IPX Routing Overview

You can route packets from your system to an external destination using the Internet Packet Exchange (IPX) protocol. The IPX protocol is a NetWare LAN communications protocol that moves data between servers and workstation programs that are running on various network nodes. IPX is a User Datagram Protocol (UDP), which is used for connectionless communications. IPX packets are encapsulated and carried by Ethernet packets and Token Ring frames.

Figure 58 shows the relationship of the IPX protocol to the Open System Interconnection (OSI) reference model.

Figure 58 IPX Protocol in the OSI Reference Model



Features

Using the IPX protocol to route packets, you can create and support:

- IPX interfaces
- IPX routes (primary and secondary)
- IPX servers (primary and secondary)
- IPX forwarding
- IPX RIP mode
- IPX SAP mode

Benefits

You can use IPX routing to:

- Provide services for connectionless communications.
- Reduce the cost of equipment moves, upgrades, and other changes and simplify network administration.
- Create VLAN-to-IPX interfaces to create virtual workgroups with most of the network traffic staying in the same IPX interface broadcast domain.
- Help avoid flooding and minimize broadcast and multicast traffic.

Key Concepts

This section explains how IPX routing works and provides a glossary of IPX routing terms.

How IPX Routing Works

To route packets using the IPX protocol, take these general steps:

1. Define an IPX protocol or a virtual LAN (VLAN) with a grouping of ports.
2. Define an IPX routing interface.
3. Decide which IPX route and server options you want to use and select these options from the Administration Console.
4. Enable IPX forwarding.

The IPX routing interface defines the relationship between an IPX VLAN and the subnetworks in the IPX network.

Each IPX VLAN interface is associated with a VLAN that supports IPX. The Multilayer Switching Module has one interface defined for each subnetwork to which it is directly connected.

A router operates at the network layer of the Open Systems Interconnection (OSI) Reference Model. The router receives instructions to route packets from one segment to another from the network-layer protocol. IPX, with the help of the Routing Information Protocol (RIP), performs network-layer tasks, including:

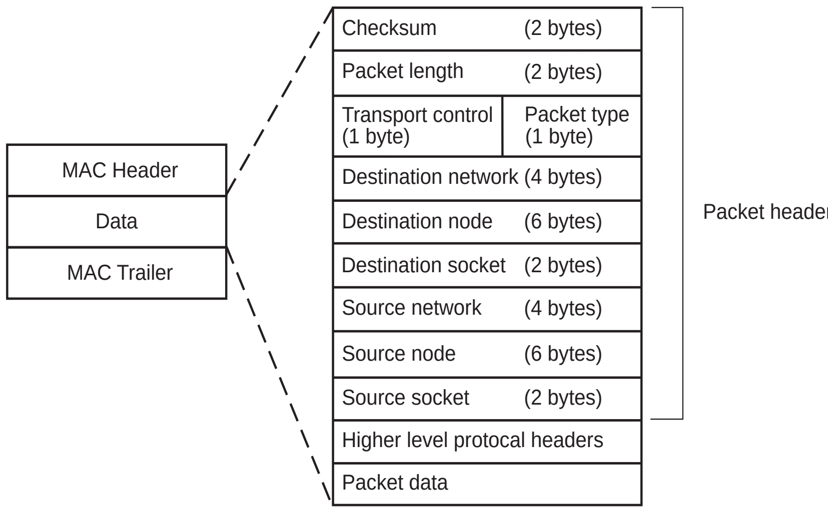
- Addressing packets
- Routing packets
- Switching packets

IPX Packet Format

An IPX packet consists of a 30-byte header followed by packet data. The packet header contains network, node, and socket addresses for both the destination and the source.

Figure 59 shows the IPX packet format.

Figure 59 IPX Packet Format



The IPX packet contains the following elements:

- **Checksum** — A 16-bit checksum that is set to 1s.
- **Packet length** — A 2-byte field that indicates the packet's length in bytes. This length includes both header and data. The length must be at least 30 bytes.
- **Transport control** — A 1-byte field that indicates how many routers a packet has passed through on its way to its destination. Packets are discarded when this value reaches 16. A network node sets this field to 0 before sending the IPX packet.
- **Packet type** — A 1-byte field that specifies the upper-layer protocol that receives the packet.
- **Destination network** — A 4-byte field that contains the network number of the destination node. When a sending node sets this field to 0, the system routes the packet as if the sending and destination nodes were on the same local segment.
- **Destination node** — A 6-byte field that contains the physical address of the destination node.
- **Destination socket** — A 2-byte field that contains the socket address of the packet's destination process.

- **Source network** — A 4-byte field that contains the source node network number. If a sending node sets this field to 0, the source's local network number is unknown.
- **Source node** — A 6-byte field that contains the source node, physical address. Broadcast addresses are not allowed.
- **Source socket** — A 2-byte field that contains the socket address of the process that transmitted the packet.
- **Packet data** — A field that contains information for upper-layer network processes.

IPX Packet Delivery

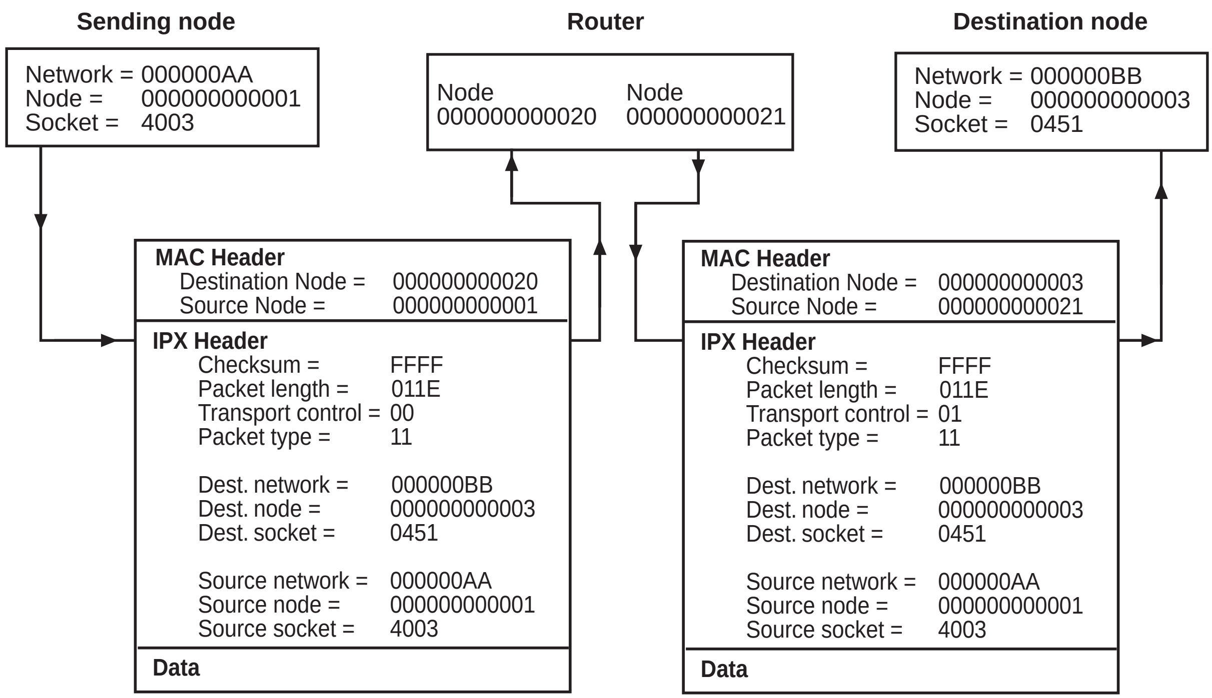
Successful packet delivery depends both on proper addressing and on the network configuration. The packet's Media Access Control (MAC) protocol header and IPX header address handle packet addressing.

The sending node must have the destination's complete network address, including the destination network, node, and socket. After the sending node has the destination address, it can address the packet.

However, the way the IPX packet's MAC header is addressed depends on whether a router separates the sending and destination nodes.

Figure 60 shows an example of IPX format routing.

Figure 60 IPX Packet Routing



Sending Node's Responsibility

When sending and destination nodes have the same network number, the sending node addresses and sends packets directly to the destination node. If sending and destination nodes have different network numbers, as in Figure 60, the sending node must find a router on its own network segment that can forward packets to the destination node's network segment.

To find this router, the sending node broadcasts a RIP packet, requesting the best route to the destination node's network number. The router on the sending node's segment that has the shortest path to the destination segment responds to the RIP request. The router's response includes its network and node address in the IPX header. After the sending node determines the intermediate router's address, it can send packets to the destination node.

If the sending node is a router rather than a workstation, the node's internal routing tables supply the destination's network location. The destination router does not need to broadcast a RIP request.

Router's Responsibility

A router handles a received IPX packet in one of two ways:

- If the packet is destined for a network number to which the router is directly connected, the sending router:
  - **a** Places the destination node address from the IPX header in the destination address field of the packet's MAC header
  - **b** Places its own node address in the source address field of the packet's MAC header
  - **c** Increases the transport control field of the IPX header by 1 and transmits the packet on the destination node segment
- If the packet is destined for a network number to which the router is not directly connected, the router sends the packet to the next router along the path to the destination node. The sending router:
  - **a** Looks up the node address in the routing information table of the next router and places the address in the destination address field of the packet's MAC header
  - **b** Places its own node address in the source address field of the packet's MAC header
  - **c** Increments the transport control field in the IPX header and sends the packet to the next router

Terminology

Review the following IPX routing terms that are used extensively throughout this chapter:

- **Address** — Unique 4-byte network address of a segment that is located in your Multilayer Switching Module's routing table.
- **Age** — The time in seconds since the network's last update.
- **Cost** — A number between `1` and `65534` that the system uses to calculate route ticks. Assign a cost of 1 to each IPX interface unless your network has special requirements like the need for redundant paths.
- **Frame formats** — Frame encapsulation format.
- **Hops** — The number of routers that must be crossed to reach a network segment.
- **Interface** — The system-assigned number for an IPX interface.
- **NetBIOS protocol** — Network Basic Input Output System protocol. An application programming interface (API) that adds special functions for PC-based LANs.
- **Node** — The node address of the router that can forward packets to each network segment (when this is set to all `0`s, the router is directly connected).
- **RIP** — Routing Information Protocol. Allows the exchange of routing information on a NetWare network. IPX routers use RIP to create and maintain their dynamic routing tables.
- **SAP** — Service Advertisement Protocol. Provides routers and servers that contain SAP mode agents with a means of dynamically exchanging network service information.
- **Ticks** — An estimate of the time that is necessary to reach a network segment. One second is 18.21 ticks.
- **VLAN interfaces** — Your Multilayer Switching Module's point of attachment to a given VLAN. A VLAN interface exists entirely within a given IPX interface.

Key Guidelines for Implementation

Consider the guidelines in this section when you configure your Multilayer Switching Module for IPX routing.

Procedural Guidelines

Complete the following steps to set up IPX routing on your Multilayer Switching Module:

1. Set up your VLAN interfaces.
2. Define the IPX interfaces before you define the routes and servers.
3. Define routes.
4. Define servers.
5. Select RIP or SAP, if you plan to use them.
6. Define IPX forwarding.

See the *Command Reference Guide* for commands that you use for these steps.

General Guidelines

Consider the following general guidelines before you configure IPX routing on your Multilayer Switching Module:

- Every IPX interface has one IPX VLAN and other associated information.
- The IPX router has one IPX interface defined for each network to which it is directly connected.
- Before you define an associated IPX interface for a network, you must first define a VLAN. See Chapter 14.
- The IPX router has one IPX interface defined for each network to which it is directly connected.

IPX Interfaces

An IPX interface has the following information associated with it:

- **IPX network address** — A 4-byte address that you assign. Make each address unique within the network.
- **Cost** — A number between 1 and 65534 that the system uses to calculate route ticks. A tick is an estimate of how long it takes a packet to reach a network segment. One tick is approximately 1/18 of a second (there are 18.21 ticks in a second). Assign a cost of 1 to each IPX interface unless your network has special requirements like the need for redundant paths.
- **Encapsulation format** — Formats that IPX routing uses: Ethernet Type II, Novell 802.3 Raw, 802.2 LLC, and 802.3 SNAP
- **State** — The status of the IPX interface. The IPX interface status can be up (available for communication) or down (unavailable for communication).
- **VLAN interface index (VLAN index)** — The VLAN that is associated with a IPX interface. When the system prompts you for this option, it indicates the available VLAN indexes.

Important Considerations

Consider the following guidelines when you set up an IPX interface:

- The first line in an interface display indicates whether:
  - IPX forwarding is enabled
  - IPX RIP mode is active
  - IPX RIP mode triggered updates are enabled
  - IPX SAP mode is active
  - IPX SAP triggered updates are enabled
  - The secondary route/server option is enabled on the system
- An IPX interface defines the relationships among an IPX VLAN, the IPX router, and the IPX network. The IPX router has one IPX interface defined for each network to which it is directly connected.

- When you define an IPX interface, you define:
  - IPX address
  - Cost
  - Format
  - Associated IPX VLAN index
- Before you define the IPX (routing) interface, you must define a VLAN and select IPX, IPX-II, IPX-802.2, IPX-802.2 LLC, or IPX-802.3-SNAP as the protocol to be supported by the VLAN. See Chapter 14 for information about creating VLANs.
- Unless your network has special requirements, such as the need for redundant paths, assign a cost of 1 to each interface and do not modify this setting.
- When you modify an IPX interface, you can change its:
  - IPX address
  - Cost
  - Format
  - Associated IPX VLAN index
- If you use the OddLengthPadding feature (10 MB switching modules support only), make sure that you select only those interfaces that require odd-length padding. If you enable this option for every interface, network performance is degraded.

For more detailed information that explains how to create an IPX interface, see the *Switch 4007 Command Reference Guide*.

Per-Interface Options

You can set the NetBIOS and OddLengthPadding options on each interface. For details about how to use these options, see the Administering IPX Routing chapter in the *Command Reference Guide*.

NetBIOS Option

This option determines whether the system handles IPX Type 20 packet forwarding.

OddLengthPadding Option

This option only supports 10 MB switching modules. To provide a compatibility mode for older network interface cards (NICs), it enables an interface to pad IPX packets that have an odd number of bytes. (Older NICs discard IPX packets that have an odd number of bytes.)

IPX Routes

Your system maintains a table of routes to other IPX networks. You can:

- Use RIP mode to exchange routing information dynamically.
- Use the Administration Console to make static entries in the table.

Important Considerations

Consider the following guidelines when you set up an IPX route:

- The first line in the output (the status line) indicates whether:
  - IPX forwarding is enabled
  - IPX RIP mode is active
  - IPX RIP mode triggered updates are enabled
  - IPX SAP mode is active
  - IPX SAP triggered updates are enabled
  - The secondary route/server option is enabled on the system
- The route table display shows the range for routing table *Primary* entries in the *N-M* format, where N is the current number of entries and *M* is the maximum number of Primary entries.
- A Secondary route entry can replace a Primary route entry when the Primary route is removed from the routing table for any reason (for example, if the route reaches its age limit).
- To view entries for any Secondary routes:
  - Establish alternate paths to the same IPX network.
  - Enable the IPX Secondary route/server option.

- The maximum number of hops, or routers that a packet can cross, is 16, except for NetBIOS packets, which can cross no more than 7 routers.
- Before you define static routes on your system, you must define at least one IPX interface.
- Static routes remain in the routing table until you remove them or until you remove the corresponding interface.
- If an interface goes down, routes are temporarily removed from the routing table until the interface comes back up.
- Static routes take precedence over dynamically learned routes to the same destination. You can have a maximum of 32 static routes.
- When you use the `ipx route remove` option to remove a route, that route is immediately removed. All servers that depend on the removed route are also removed from the Server Information Table, including all static servers.
- When you use the `ipx route flush` option to remove dynamically learned routes from the IPX routing table, all dynamically learned routes are removed immediately. All dynamic servers that depend on these routes are also removed from the Server Information Table.

Primary and Secondary Routes

You can set up both Primary and Secondary routes in the routing table.

To set up routes in the routing table, see the IPX routing chapter in the *Switch 4007 Command Reference Guide*.

Static Routes

You manually configure a static route. Static routes are useful in environments in which no routing protocol is used or in which you want to override a routing protocol's generated route.

Static routes do not change until you change them, and they do not time out. Because static routes do not change in response to network topology changes, manually configure only a small number of reasonably stable routes.

Dynamic Routes Using RIP

A router uses RIP to exchange its routing table with other routers at regular intervals. This automatic method of learning routes helps you keep up with a changing network environment and allows you to reconfigure routes quickly and reliably. Interior Gateway Protocols (IGPs), which operate within intranetworks, provide this automated learning.

The system uses RIP (one of the most widely used IGPs) to dynamically build routing tables.

RIP operates with active and passive network devices. *Active* devices, usually routers, broadcast their RIP messages to all devices in a network; they update their own routing tables when they receive a RIP message. *Passive* devices, usually hosts, listen for RIP messages and update their routing tables; they do not send RIP messages. On your system, you select a RIP mode to determine how RIP operates, as described in "IPX RIP Mode" later in this chapter.

An active router sends a RIP message every 60 seconds. This message contains both the network number for each destination network and the number of hops to reach it. In RIP, each router through which a packet must travel to reach a destination counts as one network *hop*.

Routing Tables

A routing table collects information about all intranetwork segments. This table allows a router to send packets toward their destinations over the best possible routes.

The table contains an entry for every network number that the router knows about. The router uses this information when the router is not directly connected to a packet's destination network. The routing information table provides the address of another router that *can* forward the packet toward its destination.

The routing table consists of the following elements:

- **Interface** — The interface number of the router that is used to reach a network segment
- **Address** — The network segments that the router knows about
- **Hops to network** — The number of routers that must be crossed to reach a network segment
- **Ticks to network** — An estimate of the time that it takes to reach a network segment. There are 18.21 ticks in a second.
- **Node** — The node address of the router that can forward packets to each network segment. When the node is set to all 0s, the router is directly connected.
- **Age** — The time since the network's last update

Figure 61 shows an example of a typical routing information table.

Figure 61 Sample Routing Table

Routing table

Interface	Address	Hops	Ticks	Node	Age
1	1	1	1	00-00-00-00-00-00	0
2	45469f30	1	1	00-00-00-00-00-00	0
2	45469f33	2	3	08-00-17-04-33-45	40

The routing information table is updated statically or dynamically.

Selecting the Best Route

Large networks contain many possible routes to each destination. A router performs the following steps to find the best route toward a destination:

- If one route requires the lowest number of ticks, the router selects it as the best route.
- If multiple routes require the same lowest number of ticks, the router selects the route that requires the lowest number of hops as the best route.
- If multiple routes require the same lowest number of ticks and hops, the router may select any of them as the best route.

IPX Servers

Your system creates and maintains a server information table that lists all the servers that reside on other IPX networks. You can:

- Use SAP to exchange server information dynamically
- Make static entries in the server table

Important Considerations

Consider the following guidelines when you set up an IPX server:

- The first line in the output (status line) indicates whether:
  - IPX forwarding is enabled
  - IPX RIP mode is active
  - IPX RIP mode triggered updates are enabled
  - IPX SAP mode is active
  - IPX SAP triggered updates are enabled
  - The secondary route/server option is enabled on the Multilayer Switching Module
- Static servers remain in the table until you:
  - Remove them
  - Remove the corresponding interface
  - Remove the route to the corresponding network address
- A static server must have an IPX network address that corresponds to a configured interface or to a static route. If an interface goes down, any static servers on that interface are removed from the server table until the interface comes back up.
- Static servers take precedence over dynamically learned servers to the same destination. You can have a maximum of 32 static servers.
- Before you define static servers on the system, first define at least one IPX interface.
- When you use the ipx server remove option to remove a server, that server is immediately removed from the Server Information Table.
- When you use the `ipx server flush` command to remove all dynamically learned servers, all dynamically learned servers are immediately removed from the Server Information Table.

Primary and Secondary Servers

You can set up both Primary and Secondary servers in the server table. Secondary servers serve as a backup to the Primary server set up on the same IPX server.

To set up Secondary servers on your system, see the *Switch 4007 Command Reference Guide*.

Static Servers

Static servers are useful in environments in which no routing protocol is used, or when you want to override some of the servers that generated with a routing or server protocol. Because static servers do not automatically change in response to network topology changes, manually configure only a small number of relatively stable servers.

Dynamic Servers Using SAP

Servers are automatically added to and removed from the information table through SAP. This automatic SAP update helps you keep up with changing network environments and allows servers to advertise their services and addresses quickly and reliably.

As servers boot up, they advertise their services. When servers are brought down, they use SAP to broadcast that their services are no longer available.

Client systems do not use this server information directly. Instead, SAP agents within each router on the server's network segment collect this information. The SAP agents store the information in their server information tables. Client systems then contact the nearest router or file server SAP agent to obtain server and service information.

On your system, you select a SAP mode to determine how SAP operates, as described in "IPX SAP Mode" later in this chapter.

Maintaining Server Information

When a router's SAP agent receives a SAP broadcast response indicating a change in a server's configuration, the agent updates its server information table and informs other SAP agents. Examples of such a change are when a server is disconnected or becomes accessible through a better route.

The SAP agent immediately sends an update broadcast to all directly connected network segments except the segment from which the information was received. All future periodic broadcasts contain the change information.

SAP Aging

Router SAP agents use a special aging mechanism to deal with a SAP agent that goes down suddenly without sending a DOWN broadcast. A hardware failure, power interruption, or power surge can cause this situation.

Each SAP agent maintains a timer for each entry in its server information tables. The timer tracks the elapsed time since this entry has been updated. This information is either new or changed, and the SAP agent immediately passes it on. Changes are quickly captured and stored throughout the intranetwork.

SAP Request Handling

When a SAP agent receives a general request, it notifies the sending source about all servers known to the agent. This response includes the same information that is sent out in periodic SAP broadcasts. When the request is specific, the SAP agent notifies the sending source about all servers of the requested type.

Server Tables

Server information tables contain data about all active servers on the intranetwork. SAP agents use these tables to store information received in SAP broadcasts. Server tables are dynamically and statically created.

Figure 62 shows an example of a Server Information Table.

Figure 62 Sample Server Information Table

Server information table

Interface	Name	Type	Network	Node	Socket	Hops	Age
1	LPX1102	4	45469f33	00-00-00-00-00-01	451	2	102
1	LPX1103	4	45469f44	00-00-00-00-00-01	451	5	65
2	LPX2001	4	45470001	00-00-00-00-00-01	451	4	33

This table contains the following data:

- **Interface** — The interface from which server information is received
- **Server name** — The name of the server
- **Server type** — The type of service that the server provides
- **Network address** — The address of the network that contains the server
- **Node address** — The server's node address
- **Socket address** — The socket number through which the server receives service requests
- **Hops to server** — The number of intermediate networks that must be crossed to reach the server
- **Age of server** — The time in seconds since the server's last table update

IPX Forwarding

You can control whether the system forwards or discards IPX packets with the ipx forwarding option.

Important Considerations

Consider the following guidelines before you use the `ipx forwarding` option:

- When you enable ipx forwarding, the Multilayer Switching Module acts as a normal IPX router. It forwards IPX packets from one network to another when required.
- When you disable ipx forwarding, the Multilayer Switching Module discards all IPX packets.

IPX RIP Mode

You can exchange routing information on a NetWare network with the `ipx rip mode` option. This option selects the IPX RIP mode that is appropriate for your network and selects the routers that use RIP mode to create and maintain their dynamic routing tables.

In ipx rip mode, one router exchanges routing information with a neighboring router. When a router discovers any changes in the network layout, it broadcasts this information to any neighboring routers. IPX routers also send periodic RIP broadcast packets that contain all routing information. These broadcasts synchronize all routers on the network and age those networks that might become inaccessible if a router is abnormally disconnected from the network.

Important Considerations

Consider the following guidelines before you use the `ipx rip mode` option:

- The system has three RIP modes:
  - **Off** — The system processes no incoming RIP packets and generates no RIP packets of its own.
  - **Passive** — The system processes all incoming RIP packets and responds to RIP requests, but it does not broadcast periodic or triggered RIP updates.
  - **Active** — The system processes all incoming RIP packets, responds to explicit requests for routing information, and broadcasts periodic and triggered RIP updates.
- The system has two RIP triggered modes:
  - **Disabled** — Broadcasts IPX routes 3 seconds after learning them.
  - **Enabled** — Broadcasts IPX routes immediately after learning them.

RIP Policies

Each router maintains a table of current routing information (the routing table). The routing protocols receive or advertise routes from the network. RIP policies control the flow of routing information among the network, the protocols, and the routing table manager.

Routing policies allow you to define:

- The import policies that specify which routes the router places into the routing table.
- The export policies that specify the routes that the router propagates to the network.
- Import/export policies for each peer.

RIP Import Policies

Before the router adds a route to the routing table, it follows these steps:

- The protocol receiving the route forwards the route to the routing table manager.
- The routing table manager compares the route to the import policy to determine whether to accept or drop the route.
- If the routing table manager accepts the route, it stores the route in the routing table.

The default import policy is none; that is, the router places all routes into the routing table.

RIP Export Policies

At certain times, such as when the routing table changes, the protocol asks the routing table manager for routes to advertise to other routers. The routing table manager follows these steps:

- It compares the route to the export policy to determine whether to advertise the route to the network.
- If it accepts the route, the manager propagates it to the network.

The default import policy is none; that is, the router advertises all routes in the routing table.

RIP Policy Parameters

These parameters define SAP policies:

Policy type — Import (apply the policy to received services) or Export (apply the policy to advertised services).

Route origin — The origin of the route for this policy if it is an export policy: static, RIP, or all.

Route — An IPX network address that specifies the route that applies to this policy.

Interface — One or more IP interfaces on this router that are associated with the RIP policy.

Source Node Address — The MAC address of the router that can forward packets to the network.

Action — Whether this router accepts or rejects a route that matches the policy.

Metric — Increase or decrease a route metric by a value that you specify. This parameter is valid only if the Policy Action is set to Accept (import policies).

To change the route metric of an export policy, you must adjust the metric of the import policy on the receiving router.

Weight — The metric value of this policy. This parameter specifies the order of precedence for policies that match the same route. A higher value takes precedence over a lower value.

IPX SAP Mode

IPX SAP provides routers and servers that contain SAP mode agents with a means of exchanging network service information. Through SAP, servers advertise their services and addresses. Routers gather this information and share it with other routers. With this process, routers dynamically create and maintain a database (server table) of network service information. Clients on the network determine what services are available and obtain the network address of the nodes (servers) where they can access those services. Clients require this information to initiate a session with a file server.

You determine how SAP operations on your system with the ipx sap mode option.

Important Considerations

Consider the following guidelines before you use the ipx sap mode option:

- The Multilayer Switching Module has three SAP modes:
  - **Off** — The system does not process any incoming SAP packets and does not generate any SAP packets of its own.
  - **Passive** — The system processes all incoming SAP packets and responds to SAP requests, but it does not broadcast periodic or triggered SAP updates.
  - **Active** — The system processes all incoming SAP packets, responds to explicit requests for routing information, and broadcasts periodic and triggered SAP updates.
- The system has two SAP triggered modes for updates:
  - **Disabled** — Broadcasts IPX SAP server addresses 3 seconds after learning them.
  - **Enabled** — Broadcasts IPX SAP server addresses immediately after learning them.

SAP Policies

Each router maintains a table of current configured services (the service table). SAP receives information and advertises information about the network nodes that provide these services. SAP policies control which services the router places in the service table and advertises to the network.

SAP Import Polices

Each time the router receives an advertised service, it compares the services to the import polices to decide whether to add the service to the service table or drop it. If the router accepts the service, the router adds it to the service table.

The default import policy is none; that is, the router places all services into the service table.

SAP Export Policies

At certain times, such as when a router is started up or shut down, SAP advertises services to other routers. Each time the router prepares to advertise the service, it compares it to the export policies to decide whether to advertise the service. If the export policy does not prohibit the service, the router sends it out.

The default export policy is none; that is, the router advertises all services.

SAP Policy Parameters

These parameters define SAP policies:

- **Policy type** — Import (apply the policy to received services) or Export (apply the policy to advertised services).
- **Route origin** — The origin of the service for this policy, if it is an export policy: static, SAP, or all.
- **Service type** — The Novell standard 6-digit hexadecimal number that represents the type of service offered by the server. For details, consult your Novell documentation. Refer to the *Command Reference Guide* for a list of common service types.
- **Server name** — The name of the server providing the services.
- **Network address** — The IPX network address of the network on which the server resides.
- **Node address** — The 6-byte MAC address of the router that can forward packets to the network.
- **Interfaces** — One or more IP interface index numbers associated with this policy.

- **Action** — Whether this router accepts or rejects a service that matches the policy.
- **Weight** — The metric value that is associated with this policy. This parameter specifies the order of precedence for policies that match the same service. A higher value takes precedence over a lower value.

IPX Statistics

You can view the following IPX statistics on your system:

- IPX summary statistics
- IPX RIP statistics
- IPX SAP statistics
- IPX forwarding statistics
- IPX interface statistics

In the display, the status line indicates whether:

- IPX forwarding is enabled
- RIP mode is active
- RIP mode triggered updates are enabled
- SAP mode is active
- SAP mode triggered updates are enabled
- The secondary route/server option is enabled

See the *Command Reference Guide* for more information about IPX statistics.

Standards, Protocols, and Related Reading

The following standards and protocols apply when you use IPX to route packets on your system:

- IEEE 802.2
- IEEE 802.2 LLC
- IEEE 802.3
- IEEE 802.3-RAW
- IEEE 802.3-SNAP
- Internet Packet eXchange (IPX) — RFC 1234, RFC 1552
- Routing Information Protocol (RIP) — RFC 1058
- Service Advertisement Protocol (SAP) — NetWare Protocol

21 APPLETALK ROUTING

This chapter provides guidelines and other key information about how to implement AppleTalk routing on Multilayer Switching Modules. The chapter covers these topics:

- AppleTalk Overview
- Key Concepts
- Key Implementation Guidelines
- AppleTalk Interfaces
- AppleTalk Routes
- AppleTalk Address Resolution Protocol (AARP) Cache
- AppleTalk Zones
- Forwarding AppleTalk Traffic
- Checksum Error Detection
- AppleTalk Echo Protocol (AEP)
- AppleTalk Statistics
- Standards, Protocols, and Related Reading

After you log in to the system and connect to a slot that houses a Multilayer Switching Module, you can manage AppleTalk features from the `appletalk` *menu of the Administration Console. See the* Switch 4007 Command Reference Guide.

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

AppleTalk Overview

AppleTalk is a suite of protocols defined by Apple Computer, Inc., for connecting computers, peripheral devices, and other equipment to a network. AppleTalk protocols support most of the functions that are offered by the Open Systems Interconnection (OSI) Reference Model.

The AppleTalk protocols work together to provide file sharing and printer sharing, as well as applications like electronic mail and database access. All Macintosh computers have AppleTalk connectivity options built into them, which makes it the de facto standard for Apple networks.

AppleTalk transport and application services operate over a best-effort Delivery Datagram Protocol (DDP). The AppleTalk Data Stream Protocol (ADSP) ensures reliable transmission of AppleTalk information.

Your system supports AppleTalk version 2, which runs the AppleTalk Routing Table Maintenance Protocol (RTMP). A distance-vector based routing protocol, RTMP constructs best paths based on hop-count information propagated by neighbors.

Features

AppleTalk routing includes these features:

- **AppleTalk Interfaces** — An AppleTalk interface is one that can send and receive AppleTalk traffic. When you configure an AppleTalk interface, you define the behavior and role of the interface within the AppleTalk routing domain. For example, seed interfaces propagate network configuration information, while nonseed interfaces listen for it. See "AppleTalk Interfaces" later in this chapter for more information.
- **AppleTalk Routes** — Your system maintains a table of reachable AppleTalk networks. You may want to view the contents of the table for administrative purposes. See "AppleTalk Routes" later in this chapter for more information.
- **AppleTalk Address Resolution Protocol (AARP) Cache** — The AARP cache contains a list that maps each known AppleTalk address to a corresponding MAC address. Your system lets you view this list, as well as remove entries from it. See "AppleTalk Address Resolution Protocol (AARP) Cache" later in this chapter for more information.

- **AppleTalk Zones** — All resources on an AppleTalk network are grouped into zones. Zones make AppleTalk resources easier to identify and locate. Your system maintains a zone table which maps network numbers to zones and lets you display this zone table indexed by network numbers, or by zones. See "AppleTalk Zones" later in this chapter for more information.
- **Forwarding AppleTalk Traffic** — You can disable or enable the forwarding of AppleTalk traffic on a Multilayer Switching Module. See "Forwarding AppleTalk Traffic" later in this chapter for more information.
- **Checksum Error Detection** — AppleTalk uses checksums to detect errors in data transmissions. Your system allows you to enable or disable checksum generation and verification. See "Checksum Error Detection" later in this chapter for more information.
- **AppleTalk Echo Protocol (AEP)** — Your system supports AppleTalk Echo Protocol, which you can use to test the connectivity and response of an AppleTalk device. See "AppleTalk Echo Protocol (AEP)" later in this chapter for more information.
- **AppleTalk Statistics** — You can also display AppleTalk statistics for a number of AppleTalk protocols. These statistics can help you diagnose and troubleshoot network issues and performance problems. See "AppleTalk Statistics" later in this chapter for more information.

Benefits The benefits of AppleTalk include:

- AppleTalk is built into all Apple devices, making them automatically network capable. This makes AppleTalk an extremely easy network system to install and operate.
- The naming mechanism that AppleTalk uses frees users from having to understand anything about how AppleTalk works.
- AppleTalk supports peer-to-peer networking, so no dedicated servers or centralized network control is required.
- AppleTalk is plug-and-play, or autoconfiguring. Therefore, users can plug an AppleTalk device into an AppleTalk network and use it immediately.
- No configuration of network information or assigning of network addresses is required when you add a device to an AppleTalk network.

- In theory, AppleTalk networks can support millions of nodes.
- AppleTalk supports zones, which makes it easier for network administrators to define workgroups that consist of users and services that span multiple network segments.

Key Concepts

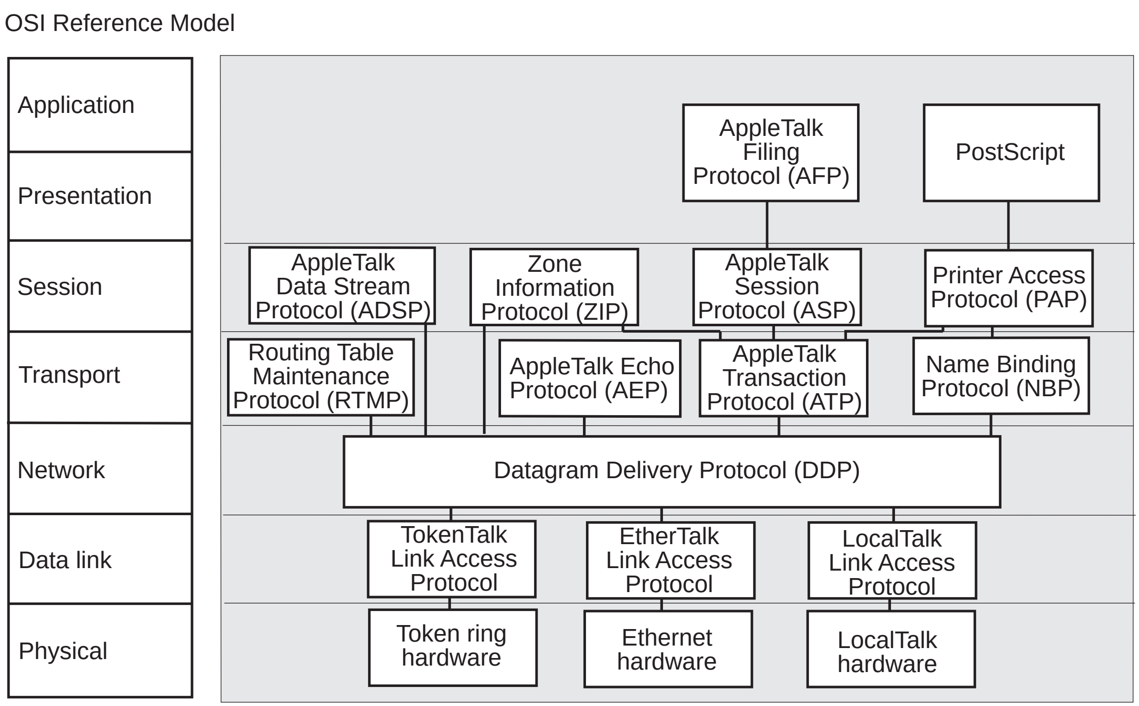
Before you configure AppleTalk, review the following key concepts and terms discussed in these sections:

- AppleTalk Protocols
- AppleTalk Network Elements
- Terminology

AppleTalk Protocols

AppleTalk protocols ensure the flow of information through AppleTalk networks. Figure 63 shows a simplified view of AppleTalk protocols and their relationship to the OSI Reference Model. These protocols provide physical connectivity, end-to-end network services, and data delivery.

Figure 63 AppleTalk Protocols and the OSI Reference Model



The AppleTalk six-layer protocol suite does not fully comply with the OSI seven-layer model. However, AppleTalk provides many of the functions and services of OSI. AppleTalk has no specific protocols for the Application layer because the lower levels provide printer and file service.

Physical Layer Protocols

The Physical layer of the OSI protocol stack defines the connection with network hardware. With AppleTalk, you can use standard network hardware, such as that designed for Ethernet and token ring networks. Apple has also defined its own network hardware, called LocalTalk, which uses a synchronous RS-422A bus for communications.

Link Layer Protocols

The data link layer provides the interface between the network hardware and the upper layers of the protocol stack. The AppleTalk data link layer includes three link access protocols (LAPs):

- TokenTalk LAP (TLAP)
- Ethernet LAP (ELAP)
- LocalTalk LAP (LLAP)

The AppleTalk Address Resolution Protocol (AARP), which translates hardware addresses to AppleTalk addresses, also exists at the data link layer because it is closely related to the Ethernet and token ring LAPs. AARP is usually included in the definition of each LAP, so it does not appear in the reference model. See "AppleTalk Address Resolution Protocol (AARP) Cache" later in this chapter for more information about this protocol.

Network Layer Protocols

The network layer accepts data from the layers above it and divides the data into packets to send over the network through the layers below it. The Datagram Delivery Protocol (DDP) transfers data in packets called *datagrams*.

Datagram delivery is the basis for building other AppleTalk services, such as electronic mail. With DDP, AppleTalk runs as a process-to-process, best-effort delivery system in which the processes running in the nodes of interconnected networks exchange packets with each other.

Transport Layer Protocols

The transport layer and the session layer provide end-to-end services in the AppleTalk network. These services ensure that routers transmit data accurately between one another. Each layer includes four protocols that work together to support these services. This section describes these protocols and provides more detail for the protocols that you can view using the Administration Console.

An AppleTalk intranet has four transport layer protocols:

- Routing Table Maintenance Protocol (RTMP)
- Appletalk Echo Protocol (AEP)
- AppleTalk Transaction Protocol (ATP)
- Name Binding Protocol (NBP)

Routing Table Maintenance Protocol (RTMP) This protocol maintains information about AppleTalk addresses and connections between different networks. It specifies that each router:

- Learns new routes from other routers
- Deletes a route if the local router has not broadcast the route to the network for a certain period of time

Each router builds a routing table for dynamic routing operations in an AppleTalk intranet. Every 10 seconds, each router sends an RTMP data packet to the network. Routers use the information that they receive in the RTMP broadcasts to build their routing tables. Each entry in the routing table contains these items:

- Network range
- Distance in hops to the destination network
- Interface number of the destination network
- State of each port (*good*, *suspect*, *bad*, or *really bad*)

A router uses these items to determine the best path along which to forward a data packet to its destination. The routing table contains an entry for each network that a router's datagram can reach within 15 hops. The table is aged at set intervals as follows:

1. After a specified period of time, the RTMP changes the status of an entry from *good* to *suspect*.
2. After an additional period of time, the RTMP changes the status of an entry from *suspect* to *bad*.
3. After an additional period of time, the RTMP changes the status of an entry from *bad* to *really bad*.
4. The router removes the entry of a nonresponding router with a *really bad* status.

The data in the routing table is cross-referenced to the Zone Information Table (ZIT). This table maps networks into zones. See "Session Layer Protocols" later in this chapter for more information about the ZIT.

Figure 64 illustrates a simple AppleTalk network, and Table 91 shows the corresponding routing table.

Figure 64 A Simple AppleTalk Network

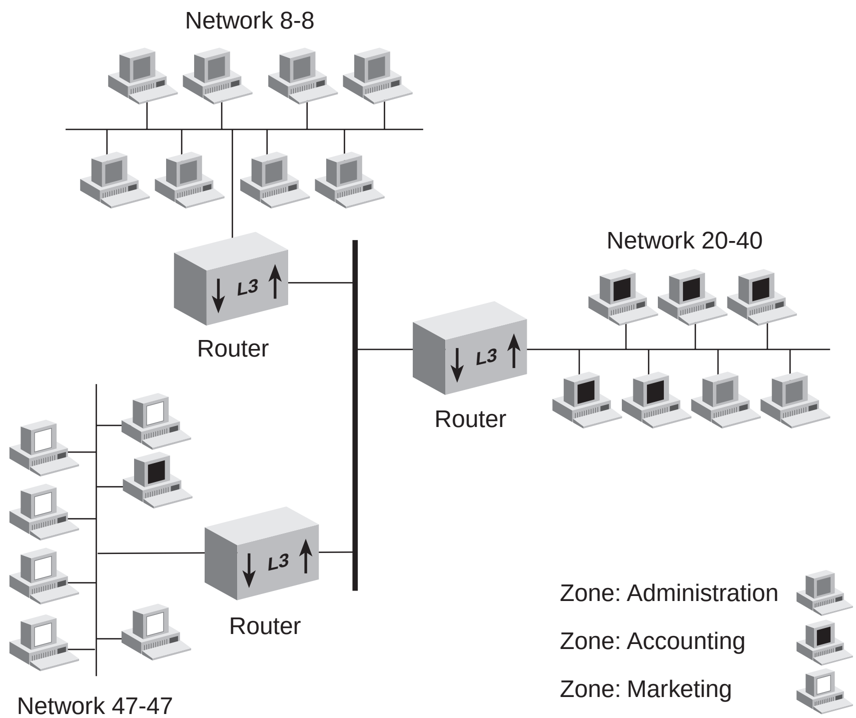


Table 91 Routing Table for Router 24 in Figure 64

Network Range	Distance (in hops)	Interface	State
5–5	1	2	Good
12–12	3	3	Good
18–20	2	3	Good
103–103	0	1	Good
64–64	1	3	Good

AppleTalk Echo Protocol (AEP) AppleTalk nodes use the AEP to send datagrams to other nodes in the network. The transmitted AEP datagram causes the destination node to return, or *echo*, the datagram to the sending node. This protocol determines whether a node is accessible before any sessions are started, and it enables users to estimate the round-trip delay time between nodes.

AppleTalk Transaction Protocol (ATP) This protocol, along with the AppleTalk Data Stream Protocol (ADSP), ensures delivery of DDP packets to a destination without any losses or corruption.

Name Binding Protocol (NBP) This protocol translates alphanumeric entity names to AppleTalk addresses. NBP maintains a table of node addresses and named entities within each node. Because each node also maintains its own list of named entities, the names directory within an AppleTalk network is not centralized. The names directory database is distributed among all nodes on the intranetwork.

Session Layer Protocols

An AppleTalk intranetwork has four session-layer protocols:

- AppleTalk Data Stream Protocol (ADSP)
- Zone Information Protocol (ZIP)
- AppleTalk Session Protocol (ASP)
- Printer Access Protocol (PAP)

AppleTalk Data Stream Protocol (ADSP) The ADSP works with the ATP to ensure reliable data transmission. Unlike ATP, however, ADSP provides full-duplex, byte-stream delivery. Therefore, two nodes can communicate simultaneously. ASDP also includes flow control, so that a fast sender does not overwhelm a slow receiver.

Zone Information Protocol (ZIP) ZIP works with RTMP to map network numbers to network zones for the entire AppleTalk intranetwork. Network zones are the logical groupings of AppleTalk networks. The table created by ZIP is called the *Zone Information Table (ZIT)*. You view the ZIT by network number or network zone from the Administration Console.

ZIP creates a zone information table in each router. Each entry in the ZIT is a *tuple*, or pair, that includes a network number and a network zone name. When an NBP packet arrives at the router, the router compares the zone name in the packet with zone names in the ZIT entries. The router then compares the network number in the matching ZIT entry with the network number in the RTMP table to find the interface for routing the packet.

AppleTalk Session Protocol (ASP) The ASP passes commands between a workstation and a server after they connect to each other. ASP ensures that the commands are delivered in the same order that they were sent and returns the results of these commands to the workstation.

Printer Access Protocol (PAP) The PAP maintains communications between a workstation and a printer or print service. The PAP functions include setting up and maintaining a connection, transferring the data, and tearing down the connection on completion of the job. Like other protocols at the session layer, PAP relies on NBP to find the addresses of named entities. PAP also depends on ATP for sending data.

Presentation Layer Protocols

The presentation layer maintains information about files, formats, and translations between formats. An AppleTalk intranetwork has two protocols at the presentation layer: the AppleTalk Filing Protocol (AFP) and PostScript. AFP provides remote access to files on the network. PostScript is a graphic page description language used by many printers.

AppleTalk Network Elements

An AppleTalk network consists of different nodes and groups of networks. Nodes can include workstations, routers, printers, and servers that provide services for other computers, called *clients*.

This section describes these elements of an AppleTalk network:

- AppleTalk Networks
- AppleTalk Nodes
- Named Entities
- AppleTalk Zones
- Seed Routers

AppleTalk Networks

A subnetwork in an AppleTalk intranetwork is a cable segment attached to a router. Each subnetwork is identified by a network number or range of network numbers. You assign these numbers from a range of valid network numbers.

Two AppleTalk network numbering systems are currently in use: nonextended (Phase 1) and extended (Phase 2). 3Com routers support extended network numbers. While the system does not translate Phase 1 packets to Phase 2 packets, it does route packets to a Phase 1 network because it anticipates that a gateway exists between the two networks to translate the packets.

An extended intranetwork can span a range of logical networks. Network numbers in an extended network consist of a range, such as network 15 through 20. This numbering scheme theoretically allows as many as 16,580,608 nodes, although the actual cables do not support this many nodes.

AppleTalk Nodes

A node in an AppleTalk network is any addressable device, including a workstation, printer, or router. Nodes are physically attached to a network. At initialization, each node in an AppleTalk network selects a unique AppleTalk address. The address consists of the node’s network number and a unique node number.

Named Entities

When a device on the network provides a service for other users, you can give the device a name. The name appears on the *Chooser* menu of the Macintosh with an associated icon. For example, the Chooser of the Macintosh can include a printer icon. When the user selects the printer icon, several printer names can appear in a list, such as `Laser1` or `Laser2`. The Name Binding Protocol (NBP), described later in this chapter, translates these device names into AppleTalk addresses.

AppleTalk Zones

An AppleTalk zone is a logical collection of nodes on an AppleTalk intranet. Zones make it easier to locate devices. Because your system supports AppleTalk, Phase 2, you can associate a list of zones for each network. Nodes on the network may belong to any of the zones associated with the network, and you can associate the same zone name with multiple networks. For more information about zones, see "AppleTalk Zones" later in this chapter.

Seed Routers

A seed router initializes the intranet with AppleTalk configuration information, including network numbers and zone names. The seed router broadcasts this information so that nonseed routers can learn it. You designate a seed router through the Administration Console.

A nonseed router listens for a seed router and takes configuration information from the first one it detects. A nonseed router that obtains configuration data participates in the network as if it is a seed router.

Terminology

If you are unfamiliar with AppleTalk routing, you may want to review the following terms:

- **Seed router** — A router that initializes the AppleTalk network with the network range and zone list information that you configure.
- **Non-seed router** — A router that listens for a seed router and obtains its network range and zone information from the seed interface that it detects.
- **Zone** — A logical subset of the systems on an AppleTalk internetwork, for example, a logical group of AppleTalk networks. For more information, see "AppleTalk Zones" later in this chapter.
- **Network range** — The range of network numbers that are assigned to an AppleTalk extended (Phase 2) network.

- **Phase 1 network** — An AppleTalk network that contains a single network number (such as network 2). Also known as *non-extended networks*, Phase 1 networks do not allow two nodes on a single network segment to belong to different zones.
- **Phase 2 network** — An AppleTalk network that contains multiple consecutive network numbers (such as network 3–20). Also known as *extended networks*, Phase 2 networks can also contain a single network number (such as network 3–3).
- **AppleTalk Address Resolution Protocol (AARP)** — An AppleTalk support protocol that maps the hardware address of an AppleTalk node to an AppleTalk protocol address.
- **Hop Count** — The number of routers that a packet must cross to reach a destination network.
- **AppleTalk Echo Protocol (AEP)** — An AppleTalk support protocol used to test the accessibility of a system and make an estimate of the route-trip transmission time that is required to reach the system.
- **Checksum** — A method providing error detection for AppleTalk packets, calculated by summing a set of values.

Key Implementation Guidelines

Consider the following guidelines when designing a dependable and scalable AppleTalk network:

- All AppleTalk routers on the same network segment must have the same configuration. Therefore, all seed routers must be configured with matching:
  - Network ranges
  - Default zones
  - Zone lists

  If a configuration mismatch occurs between routers on the same segment, then unpredictable behavior may result. For example, zones may fail to appear in Chooser, and AppleTalk services may become inaccessible.
- If you are connecting your system's AppleTalk Phase 2 routing interface to an AppleTalk Phase 1 network, follow these guidelines:
  - Specify a network range of 1 (for example, `22-22`).
  - The network can belong to only one zone.

AppleTalk Interfaces

On the Switch 4007, an AppleTalk interface defines the relationship between a virtual LAN (VLAN) and an AppleTalk network. An AppleTalk interface has these elements associated with it:

- **Seed Interface** — You can configure the interface to be a seed or nonseed interface:
  - *A seed interface* initializes (“seeds”) the network with your configuration information. This information includes the network range and zone name list.
  - *A nonseed interface* listens for a seed router and then takes the zone and network range information from the first seed interface that it detects. After a nonseed interface obtains this information, it can participate in AppleTalk routing.
- **Network Range** — The contiguous range of numbers assigned to the interface (for example, 20301 through 20310). Each router attached to the network selects a network number from within this range.
- **Address** — The AppleTalk interface address, which is based on the network range and a unique network node number (*1* through *253*) and expressed in the format *network.node*. The network number identifies the network. The node number uniquely identifies the AppleTalk node on the network. The router selects the network number from the range of numbers assigned to the network. It then selects an available node number. Sample interface address: *20301.7.*
- **Zone List** — The zone or zones to which the interface belongs. You specify the default zone name and up to 15 additional zones, for a maximum of 16 zones per interface.
- **State** — The status of the AppleTalk interface, which indicates whether the interface is available (*enabled*) or unavailable (*down*).
- **VLAN interface index (VLAN index)** — The VLAN that is associated with the AppleTalk interface. When the system prompts you for a VLAN interface index, it indicates the available VLANs that you can associate with a new AppleTalk interface. For information on creating VLANs, see Chapter 14.

Important Considerations

Before you configure AppleTalk interfaces, review the following guidelines and considerations:

- Your system can support up to 32 AppleTalk interfaces.
- Each seed interface supports up to 16 zones.
- Your system supports a maximum of 1 AppleTalk interface per VLAN; overlapping AppleTalk interfaces on a bridge VLAN are not allowed.
- A seed router interface maintains its configuration (local zone and local network information) even if the information conflicts with other routers on the same network.
- The network range is a contiguous range of numbers between 1 and 65,279.
- The network node number that a router dynamically assigns to itself is a value between 1 and 253, inclusive.
- Node numbers 0, 254, and 255 are reserved by the AppleTalk protocol.
- The maximum number of active AppleTalk devices on a network is equal to the number of network numbers multiplied by the number of possible node numbers.
- All seed routers on a particular network must have the same value for both the start and end of the network number range. For example, if you have a segment to which multiple routers are attached and you have assigned a network range of 4–9, then all seed router ports attached to the segment must also be configured with a network range of 4–9.
- All seed routers on a particular network must be configured with the same zone names. For example, if you have a segment to which multiple routers are attached and you have assigned the zone names *Sales* and *Marketing* to the segment, then all seed routers attached to the segment must also be configured with the zone names *Sales* and *Marketing*.

- A router does not advertise its routing table through an interface until that interface has an associated network number range.
- An interface is not added to the routing table until it has an associated network number range.

Changing the zone association for an existing network number involves the deletion of the existing zone association for that network from all routers on the segment. For details, see "Changing Zone Names" later in this chapter.

AppleTalk Routes

Your system maintains a table of local and remote routes to all reachable AppleTalk networks. The Routing Table Maintenance Protocol (RTMP) automatically generates the routing table. RTMP defines rules for:

- **Information contained within each routing table** — Routers use the information within this table to determine how to forward data on the basis of its destination network number.
- **Exchanging information between routers so that the routers can maintain their routing tables** — All AppleTalk routers periodically exchange routing tables by broadcasting RTMP packets onto the network every 10 seconds; each packet contains a router's routing table entries. When a router receives the routing table of another router, it compares its own table to the one it received and then updates its table with the shortest path to each destination network.

Each routing table entry contains the following information:

- **Network Range** — A range of 16 bit numbers that identifies a network. Each device on the network selects from this range the network number with which it identifies itself on the network.
- **Distance** — Number of hops to the destination network
- **Interface** — Interface used to reach the destination network
- **State** — Status (*good, suspect, bad,* or *really bad*) of each route
- **Next Hop** — The next-hop Internet router to which the packet must be sent

Important Considerations

Before administering AppleTalk routes, review the following guidelines and considerations:

- The RTMP table supports a maximum of 514 entries.
- AppleTalk supports a maximum distance of 15 hops.
- A hop count of 0 represents a network that is directly connected to the router.
- When an AppleTalk router starts up on the network, the first entries in its routing table are the network numbers to which it is directly attached.
- Node numbers are dynamically assigned and often change when the router restarts.
- Each 16 bit number within a network range is capable of supporting 253 network nodes.
- When a router receives an RTMP packet that contains a routing entry currently not in its table, the router adds the entry to its routing table, and increments the route's distance (hop count) by 1.
- When a network is removed from the RTMP table (whether manually or though the aging process), the router also scans the Zone Information Table (ZIT), and removes ZIT entries that contain the deleted network number.
- If the Zone Information Table contains an entry whose network number range is not in the RTMP table, the router then concludes that the network is no longer on the Internet and deletes the network's ZIT entry.
- An overburdened network with many routers can prevent some routers from sending RTMP updates every 10 seconds. Because routers begin to age out routes after the loss of 2 successive RTMP updates, the failure for RTMP packets to arrive may result in unnecessary route changes, known as *route flapping*. For this reason, keep network segments to a reasonable size.

AppleTalk Address Resolution Protocol (AARP) Cache

The AppleTalk Address Resolution Protocol (AARP) maps the hardware address of an AppleTalk node to an AppleTalk protocol address. AARP maps addresses for both extended and nonextended networks.

AppleTalk uses dynamically assigned 24-bit addresses that consist of a 16-bit network number and a unique 8-bit node number. AppleTalk networks support a hierarchal addressing scheme in the form of a network range, with each 16-bit network number within that range capable of supporting up to 254 nodes.

All AppleTalk nodes, including router interfaces, dynamically acquire a unique AppleTalk address using a feature provided by the AppleTalk Address Resolution protocol, called Probe.

When a node on the network initializes, it randomly selects an AppleTalk address for itself. At the same time, the node sends 10 AARP probe packets. The probe packets determine whether any other nodes on the network are using the selected address. If the address already exists, the initializing node randomly selects another address and sends another set of probe packets.

AARP maintains an Address Mapping Table (AMT) with the most recently used hardware addresses and their corresponding AARP addresses. If an address is not in this table, the router broadcasts AARP requests to all other AppleTalk nodes on the link to determine the MAC address mapping for the specified AARP address. The router then creates a corresponding AMT entry to reflect the new mapping when the destination node replies. You can view this table, called the *AARP cache*, through the Administration Console.

AARP registers a node's dynamically assigned address on the network, as follows:

- AARP randomly assigns an address.
- To determine whether another node is already using the address, the system broadcasts AARP probe packets containing the address.
  - If the system receives no reply, the address becomes the node's address.
  - If the system receives a reply, it repeats the process until it discovers an available address.

AARP entries include the following information:

- **AARP address** — AARP address of the node in *network.node* format
- **MAC address** — MAC layer address of the node
- **Interface** — Interface through which the node can be reached
- **Age** — Number of seconds before the system ages out the cache entry

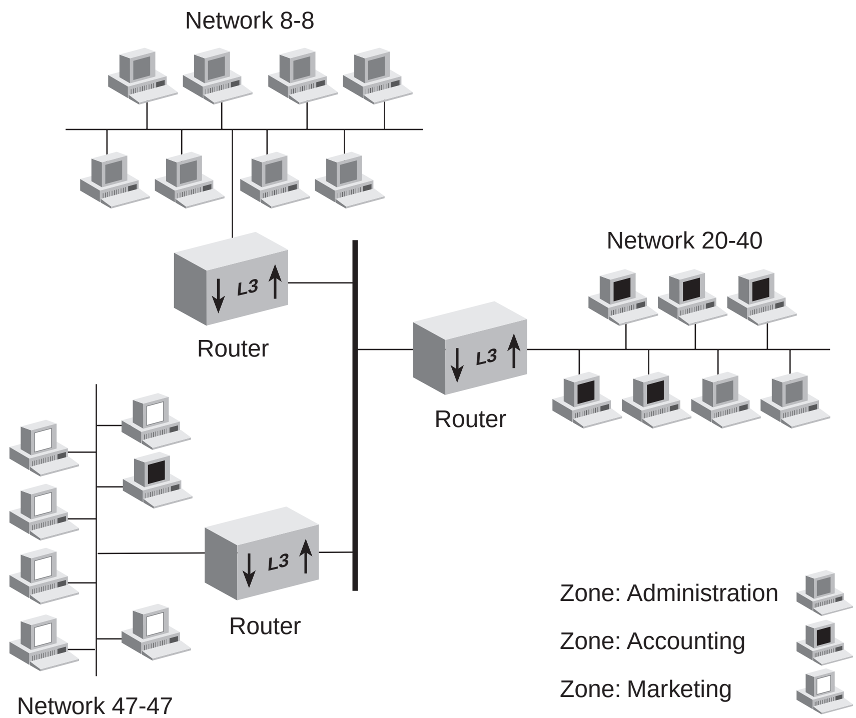
If there is no space available in the AARP cache for a new entry, the least recently used entry is purged to make room for the new entry.

AppleTalk Zones

An AppleTalk zone is a logical collection of nodes on an AppleTalk intranetwork. A zone can include all nodes in a single network or a collection of nodes in different networks. You assign a unique name to each zone to identify it in the intranetwork.

Figure 65 illustrates the relationship between physical AppleTalk networks and logical AppleTalk zones.

Figure 65 AppleTalk Networks and Zones



This example shows an AppleTalk intranet with three subnetworks: 47-47, 20-40, and 8-8. Three AppleTalk zones span these networks: Administration, Accounting, and Marketing. Network 20-40 includes two nodes in the Administration zone and five nodes in the Accounting zone. Network 47-47 includes a node from the Accounting zone and all nodes in the Marketing zone. Network 8-8 consists of nodes in the Administration zone only.

AppleTalk routers use the Zone Information Protocol (ZIP) to map network numbers to zones. Each AppleTalk router maintains a Zone Information Table (ZIT), which lists the zone-to-network mapping information.

Creating zones within a network reduces the amount of searching that a router must do to find a resource on the network. For example, to gain access to a printer on the network, instead of searching the whole network when you want to print a file to a certain printer, the router searches for it within a particular zone. You gain access to the printer more quickly within the zone because the zone includes fewer devices than the entire intranet.

Important Considerations

Before administering zones, review the following guidelines and considerations:

- Whenever a router discovers a new network, it adds the network to its RTMP table. It then creates a corresponding ZIT entry with a zone list of NIL. The ZIP process then requests from the originating router the corresponding zones associated with the newly discovered network. When ZIP receives the associated zones, it then updates the ZIT entry.
- If the Zone Information Table contains an entry whose network number range is not in the RTMP table, the router then concludes that the network is no longer on the Internet, and deletes the network's ZIT entry. This means, whenever a network is removed from the RTMP table (whether manually, or though the aging process), the router also removes ZIT entries that contain the deleted network number.
- At the time of initialization, the Zone Information Table contains an entry for each seed interface that is directly connected to an AppleTalk network.
- On a stable AppleTalk network, the ZIP process only occurs when a new router (new network number) is introduced to the network. ZIP traffic at any other time can be an indication of network instability.
- Whenever a network is aged-out and removed from the routing table, the corresponding zone information for that network is removed from the router's Zone Information Table.
- Assign zone names that are meaningful for end users.

Changing Zone Names

When you change the zone information for a network, all routers on the segment must update their Zone Information Tables with the new information. Although no AppleTalk mechanism forces routers to update zone lists, you can successfully change the zones that are associated with a network segment by aging out the existing network range, as described in the following section.

Aging Out the Network Range

If you want to change the zone information for a segment and retain the existing network range, you must age out the range from all routers on the network. This practice ensures that all routers query for the new zone information. This is necessary because after a zone has been acquired, routers do not query for zone information until the network has been aged out of their routing tables.

If you do not age out the network range, some routers may not remove the network from their routing tables. Devices attached to these networks are then unaware of the new zone information, which can result in some users seeing the new zones in their Choosers, while others see the old zones.

To age out the network range, you must prevent routers on the network from sending RTMP messages that contain the network range for a minimum amount of time, known as the ZIP bringback time. ZIP defines a bringback time of 10 minutes.

During this time, all AppleTalk interfaces on the network segment are brought down and cannot send or receive RTMP packets to confirm the existence of the network in their RTMP tables. The unconfirmed network ranges are then aged out of their routing tables; the associated zone information for the network range is removed as well.

To change the associated zones for a network segment without changing the segment's network range, follow these steps:

1. Remove all AppleTalk interfaces attached to the segment for which you want to redefine zone information.
2. Wait a minimum of 10 minutes while routers on the internetwork age out the existing network information.
3. Redefine all AppleTalk interfaces with the new zone information.
4. Start the seed routers.
5. Start the nonseed routers.

Although ZIP defines the minimum down time of 10 minutes, the exact time required to ensure that the network range is aged from all routers depends on the complexity and size of the network to which your AppleTalk segment is attached.

Forwarding AppleTalk Traffic

You can choose to enable or disable AppleTalk forwarding on your system.

Enabling Forwarding

When you enable AppleTalk forwarding, you enable the forwarding of Datagram Delivery Protocol (DDP) packets. Because AppleTalk uses this network layer protocol, this setting also enables the routing of AppleTalk packets. This means AppleTalk interfaces can forward routable AppleTalk traffic. All nonroutable protocols, or protocols not yet configured for routing, are dropped.

Disabling Forwarding

When you disable AppleTalk forwarding, you disable the forwarding of Datagram Delivery Protocol (DDP) packets. Because AppleTalk uses this network layer protocol, this setting also disables the routing of AppleTalk packets. This means that AppleTalk interfaces do not forward routable AppleTalk traffic. All AppleTalk traffic is dropped. In addition, all traffic from nonroutable protocols, or protocols not yet configured for routing, are dropped.

Important Considerations

Consider the following guidelines when you enable or disable AppleTalk forwarding:

- AppleTalk forwarding is *disabled* by default.
- The requirement that you must specifically enable AppleTalk forwarding makes it possible for you to verify that you have correctly set all necessary AppleTalk configuration parameters before you activate AppleTalk routing.

Checksum Error Detection

You can enable or disable checksum generation and verification. The AppleTalk protocol uses checksums to detect errors in data transmissions. A *checksum* totals all data bytes and adds the sum to the checksum field of the data packet. The receiving station computes a verification checksum from the incoming data and compares the new checksum with the value sent with the data. If the values do not match, the transmission contains an error.

Important Considerations

Before you configure checksum error detection, review the following guidelines and considerations:

- By default, checksum generation and verification is *disabled*.
- *Disabled* is the preferred setting. Enabling the checksum generation or verification significantly impacts the router's performance.
- You may want to disable checksum generation and verification if you have older devices that cannot receive packets that contain checksums.

AppleTalk Echo Protocol (AEP)

The system supports the AppleTalk Echo Protocol, which sends a datagram (an Echo Request) to a specified node. The destination node returns, or *echoes*, the datagram to the sender (using an Echo Reply). This process allows you to determine whether a node is accessible. Your system's `appletalk ping` command is equivalent to an IP ping, except that you specify an AppleTalk address instead of an IP address. Use this command to verify whether or not an AppleTalk node is reachable from the router.

AppleTalk Statistics

You can view statistics for the following AppleTalk protocols:

- Datagram Delivery Protocol (DDP)
- Routing Table Maintenance Protocol
- Zone Information Protocol
- Name Binding Protocol

Datagram Delivery Protocol (DDP)

AppleTalk extends the normal node-to-node delivery of packets to a process-to-process delivery. The processes running on AppleTalk nodes exchange data packets through logical sockets that are assigned by the Datagram Delivery Protocol. DDP provides a best-effort, socket-to-socket delivery of datagrams — packets exchanged using DDP — over the AppleTalk network.

Datagram delivery is the key service on which other AppleTalk services are built. All other AppleTalk services, such as RTMP, NBP, and ZIP, rely on DDP for packet delivery, as illustrated in Figure 63 earlier in this chapter.

Your system allows you to view a variety of DDP statistics, including:

- **inBcastErrors** — Number of dropped DDP datagrams for which the system was not their final destination and they were sent to the broadcast MAC address
- **inCsumErrors** — Number of DDP datagrams that were dropped because of a checksum error
- **inDiscards** — Number of DDP Datagrams that were discarded during routing
- **inForwards** — Total number of packets that were forwarded, including those with errors
- **inLocals** — Number of DDP datagrams for which an attempt was made to forward them to their final destination
- **inNoClients** — Number of DDP datagrams that were dropped for unknown DDP types
- **inNoRoutes** — Number of DDP datagrams that were dropped for unknown routes
- **inReceives** — Total number of packets that were received, including those with errors

- **inShortDdps** — Number of input DDP datagrams that were dropped because the system was not their final destination and their type was short DDP
- **inTooFars** — Number of input datagrams that were dropped because the system was not their final destination and their hop count would exceed 15
- **inTooLongs** — Number of input DDP datagrams that were dropped because they exceeded the maximum DDP datagram size
- **inTooShorts** — Number of input DDP datagrams that were dropped because the received data length was less than the data length specified in the DDP header, or the received data length was less than the length of the expected DDP header
- **outLocals** — Number of host-generated DDP datagrams

Routing Table Maintenance Protocol

AppleTalk uses the Routing Table Maintenance Protocol (RTMP) to build and maintain routing tables. Your system allows you to view a variety of RTMP statistics, including:

- **inDatas** — Number of good RTMP data packets that were received
- **inOtherErrs** — Number of RTMP packets received that were rejected for an error other than a version mismatch
- **inRequests** — Number of good RTMP request packets that were received
- **inVersionErrs** — Number of RTMP packets received that were rejected due to a version mismatch
- **outDatas** — Number of RTMP data packets that were sent
- **outRequests** — Number of RTMP request packets that were sent
- **routeDeletes** — Number of times that RTMP deleted a route that was aged out of the table
- **routeEqChgs** — Number of times that RTMP changed the Next Internet Router in a routing entry because the hop count advertised in a routing table was equal to the current hop count for a particular network

- **routeLessChgs** — Number of times that RTMP changed the Next Internet Router in a routing entry because the hop count advertised in a routing table was less than the current hop count for a particular network
- **routeOverflows** — Number of times that RTMP attempted to add a route to the RTMP table but failed because of lack of space

Zone Information Protocol

AppleTalk uses the Zone Information Protocol (ZIP) to maintain a mapping between networks and zone names. This network-to-zone mapping is used to facilitate the name-lookup process performed by the Name Binding Protocol. Your system allows you to view a variety of ZIP statistics, including:

- **inErrors** — Number of ZIP packets that have been received and rejected for any error
- **inExReplies** — Number of ZIP extended replies that have been received
- **inGniReplies** — Number of ZIP GetNetInfo reply packets that have been received
- **inGniRequests** — Number of ZIP GetNetInfo request packets that have been received
- **inLocalZones** — Number of Zip GetLocalZones requests packets that have been received
- **inObsoletes** — Number of ZIP Takedown or ZIP Bringup packets that have been received
- **inQueries** — Number of ZIP queries that have been received
- **inReplies** — Number of ZIP replies that have been received
- **inZoneCons** — Number of times that a conflict has been detected between this system's zone information and another entity's zone information
- **inZoneInvs** — Number of times that this system has received a ZIP GetNetInfo reply with the zone invalid bit set because the corresponding GetNetInfo request had an invalid zone name
- **inZoneLists** — Number of Zip GetZoneLists requests packets that have been received
- **outAddrInvs** — Number of times that this system had to broadcast a ZIP GetNetInfo reply because the GetNetInfo request had an invalid address

- **outExReplies** — Number of ZIP extended replies that have been sent
- **outGniReplies** — Number of ZIP GetNetInfo reply packets that have been sent out of this port
- **outGniRequests** — Number of ZIP GetNetInfo packets that have been sent
- **outLocalZones** — Number of transmitted ZIP GetLocalZones reply packets
- **outQueries** — Number of ZIP queries that have been sent
- **outReplies** — Number of ZIP replies that have been sent
- **outZoneInvs** — Number of times that this system has sent a ZIP GetNetInfo reply with the zone invalid bit set in response to a GetNetInfo request with an invalid zone name
- **outZoneLists** — Number of transmitted ZIP GetZoneList reply packets

Name Binding Protocol

AppleTalk uses the Name Binding Protocol (NBP) to convert user-friendly entity names (which are user-defined and change infrequently) into AppleTalk network addresses (which are dynamically assigned and change frequently). Your system allows you to view a variety of NBP statistics, including:

- **inBcastReqs** — Number of NBP Broadcast Requests that have been received
- **inErrors** — Number of NBP packets that have been received and rejected for any error
- **inFwdReqs** — Number of NBP Forward Requests that have been received
- **inLkupReplies** — Number of NBP Lookup Replies that have been received
- **inLkupReqs** — Number of NBP Lookup Requests that have been received

Standards, Protocols, and Related Reading

For more information about AppleTalk technology, see the following publications:

- Gursharan S. Sidhu, Richard F. Andrews, and Alan B. Oppenheimer, *Inside AppleTalk,* Second Edition (Addison-Wesley Publishing Company, 1990).
- RFC 1742, AppleTalk Management Information Base II

22 QOS AND RSVP

This chapter provides guidelines and other key information about how to use Quality of Service (QoS) and the Resource Reservation Protocol (RSVP) on a Multilayer Switching Module. The chapter covers these topics:

- QoS Overview
- Key Concepts
- Key Guidelines for Implementation
- QoS Classifiers
- QoS Controls
- Examples of Classifiers and Controls
- Modifying and Removing Classifiers and Controls
- QoS Excess Tagging
- Transmit Queues and QoS Bandwidth
- RSVP

You can manage QoS features from the `qos` *menu of the Administration Console. (See the* Switch 4007 Command Reference Guide.) *You can use the Administration Console after you log in to the system and connect to a slot that houses a Multilayer Switching Module.*

The management interfaces display "`cb9000`*" and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

QoS Overview

Quality of Service (QoS) is a Layer 3 feature that allows you to establish control over network traffic. QoS provides *policy-based services*, which establish various grades of network service to accommodate different types of traffic, such as multimedia, video, protocol-specific, time-critical, and file-backup traffic. Although QoS and Class of Service (CoS) are closely related, QoS has more features and addresses bandwidth, delay, loss, and jitter control. (CoS tends to focus on differentiating traffic into classes and assigning prioritization to those classes.)

QoS is crucial in the wide area network (WAN) environment to guarantee quality of service without escalating WAN bandwidth costs. In the LAN environment, QoS implementations are growing.

Features

The Multilayer Switching Modules that are available on the Switch 4007 support the following QoS features:

- **QoS Classifiers** — Define how the Multilayer Switching Module groups packets to schedule them with the appropriate service level.
- **QoS Controls** — *A*ssign rate limits and IEEE 802.1p priorities, as well as prioritize packets that are associated with one or more classifiers. Using the QoS Excess Tagging feature, you can also select an IEEE 802.1p priority for packets that exceed the control's rate limit.
- **Settable QoS Bandwidth** — Controls the weighting of high-priority and best-effort traffic.
- **Resource Reservation Protocol (RSVP)** — A building block of QoS that implements QoS characteristics in your LAN environment. RSVP is an end-to-end signaling IP protocol that allows an end station to request the reservation of bandwidth across the network. RSVP provides admission control. QoS can operate at Layer 2 and Layer 3; RSVP operates at Layer 3 only.

Benefits

You can use QoS on your Multilayer Switching Module to provide the following benefits:

- Control a wide variety of Ethernet network traffic by:
  - Classifying traffic based on packet attributes such as protocol type, class type (802.1p), IP address, or TCP/UDP socket
  - Assigning priorities to traffic (for example, to set higher priorities for time-critical or business-critical applications)
  - Applying security policy through traffic filtering
  - Using the connection-oriented RSVP for bandwidth reservation (reserving and policing an RSVP session to make sure the session uses only as much bandwidth as it needs)
- Provide constant delay control and jitter control for multimedia applications such as video conferencing or voice over IP
- Improve performance for specific types of traffic and preserve performance as the volume of traffic grows
- Reduce the need to constantly add bandwidth to the network
- Manage network congestion

Methods of Using QoS

Your Multilayer Switching Module's implementation of QoS focuses on traffic classification, policy-based management, and bandwidth. It provides multiple service levels (mapped to several transmit queues), classification of traffic types, and weighted fair queueing of priority-queued traffic.

If you use QoS and decide to classify traffic broadly, you are using a subset of QoS called *network class of service*. To simplify your classification of traffic, the Multilayer Switching Module provides a set of predefined traffic classes. You can also specify your own classes of traffic with applied controls to:

- Create a to/from classifier with address/port patterns that isolate traffic based on source and destination.
- Block traffic (for example, prevent certain traffic from one workgroup from seeing another workgroup).
- Assign priorities to traffic.

See "Examples of Classifiers and Controls" later in this chapter.

If you use QoS with RSVP, you are opting for a more complex type of end-to-end QoS that aims for a *guaranteed* quality of service. To use RSVP, you must be routing. In addition, RSVP is required at the desktop, which may present issues of desktop control and upgrade issues concerning the resident operating-system and applications.

Key Concepts

Before you configure QoS, review the following standards and terms.

Related Standards and Protocols

The Switch 4007 Multilayer Switching Modules support IEEE 802.1Q, IEEE 802.1p, and the RSVP protocol.

IEEE 802.1p

This finalized standard, part of IEEE 802.1D, covers traffic class and dynamic multicast filtering services in bridged LANs. It uses the same tag format as the proposed IEEE 802.1Q standard, but it uses three additional bits of the tag control information to set a user priority level (for policy-based services such as QoS). You can classify traffic using a specific IEEE 802.1p priority tag value (or several tag values). You can also define a control that inserts a priority tag value in forwarded frames.

The IEEE 802.1p priority tag values are 0 through 7. Table 92 shows the IEEE 802.1p (user-priority) values and the corresponding traffic types. The value 7 (Network Control) is the highest priority and 1 (Background Traffic) is the lowest priority. The value 0 (the default, Best Effort) has a higher priority than value 2 (Standard).

Table 92 IEEE 802.1p User Priority Tag Values

Tag Value	Traffic Type
1	Background
2	Standard (spare)
0 (the default)	Best Effort
3	Excellent Effort (Business Critical)
4	Controlled Load (Streaming Multimedia)
5	Video (Interactive Media), less than 100 milliseconds latency and jitter
6	Voice (Interactive Voice), less than 10 milliseconds latency and jitter
7	Network Control (Reserved Traffic)

The IEEE 802.1p standard addresses separate queuing of time-critical frames to reduce the jitter that is caused by multicast flooding. This standard also defines the Generic Attribute Registration Protocol (GARP), a Layer 2 transport mechanism that allows switches and end systems to propagate information across the switching domain.

Resource Reservation Protocol (RSVP)

This connection-oriented IP protocol handles bandwidth reservation. The request for comments document, RFC 2205, describes the details of RSVP.

RSVP aims to meet the demands of real-time voice and video applications with its QoS flow specification, which mandates parameters such as the maximum frame transmission rate, long-term average frame transmission rate, maximum frame jitter, and maximum end-to-end delay. RSVP supports the QoS flow specifications by managing *resource reservations* across the network.

With RSVP, all devices in the path from the source to the destination must agree to observe the RSVP call request parameters before traffic can flow.

Terminology

The following terms apply to QoS:

- **Classifiers** — Two types of classifiers define how your Multilayer Switching Module groups packets in order to schedule them with the appropriate service level:
  - **Flow classifiers** — Apply to *routed* IP unicast and IP multicast traffic only, not bridged traffic. (When the Multilayer Switching Module is bridging, you cannot classify to the IP address or socket level.) These classifiers are numbered in a range of from *1* through *399.* To define filtering parameters for a flow classifier, set the source IP address, source IP address mask, the destination IP address, destination IP address mask, and the TCP or UDP source and destination port range. Because these classifiers have lower class numbers, they take precedence over nonflow classifiers. When a packet falls into more than one controlled classifier, the Multilayer Switching Module uses the lower-numbered classifier to classify the packet. A Multilayer Switching Module predefines two flow classifiers for you: Telnet and FTP.

- **Nonflow classifiers** — Apply to *both switched and routed* traffic. You define this type of classifier to handle specific link-level protocols *(IP, TCP/IP, IPX,* or *AppleTalk)*, a *cast* type *(broadcast, unicast,* or *multicast)*, and one or more IEEE 802.1p priority tag values. Nonflow classifiers are numbered in a range of from *400* through *499.* The Multilayer Switching Module automatically defines a number of nonflow classifiers for you. The predefined nonflow classifiers *(401* through *407)* employ IEEE 802.1p tagging by default for received frames.

  You can configure QoS nonflow classifiers to prioritize or filter based on IP, IPX, and AppleTalk protocols; Ethertype values; or DSAP/SSAP values.

  You can also specify starting and ending ranges for source and destination ports when you define classifiers using TCP and UDP protocols. Specifying a small port range lets you limit the amount of classified traffic on the system. These port range choices are shown in the `qos detail` display.

- **Controls** — Define the following parameters to assign rate limits and priorities to the packets that are associated with one or more classifiers:

  - **Rate limit** — Limits the amount of input bandwidth used by incoming classified traffic (optionally, on a per-port basis). When you define a control, you can specify one of three rate limits: `none` (no rate limit), `receivePort` (a separate limit on each specified receive port), or `aggregate` (limits on groups of receive ports).

  - **Service levels** — Specify a transmit priority and map to a specific transmit queue. If you specify `receivePort` or `aggregate` for a rate limit, you can assign a service level of `high`, `best`, or `low` both to conforming packets (packets that are below the rate-limit parameters) and to nonconforming excess packets (excess packets that exceed the rate-limit parameters). If you set the rate limit to `none`, you can specify a service level of `high`, `best`, `low`, or `drop` for conforming classified packets.

  - **Time of day controls —** A QoS timer option enables you to configure a QoS control session. You set a start time and an end time for the specific control. After it is set, the control that is associated with this time setting becomes active if the current time is within the range of start and end time. You can preset a time period to activate the control.

Drop causes the Multilayer Switching Module to drop all packets on all ports that are associated with the control and its classifier. To drop conforming packets for only a subset of ports, specify the receivePort or aggregate rate limit, set the rate limit to 0, and specify the group of ports.

- **Loss-eligible status** — *Loss-eligible packets* are conforming packets that are discarded instead of queued when transmit queues back up beyond a threshold. You can specify whether conforming packets (as well as nonconforming excess packets) are loss eligible when you define a control. Marking packets as loss eligible helps create an intelligent discard of traffic in a congestion situation. When the Multilayer Switching Module is congested, you can decide which traffic can be discarded and mark that traffic as loss eligible.
- **Burst size** — The maximum amount of data that you can transmit at the line rate before the transmission is policed. This value accommodates variations in speeds and allows you to occasionally exceed the configured rate.
- **TCP drop control —** TCP drop control allows you to create QoS Flow Classifiers that allow traffic going from *source* IP addresses to *destination* IP addresses to be dropped or otherwise controlled using one-way TCP flow filtering. This control can only be used for flow classifiers that use the TCP/IP protocol.
- **Timer option —** The QoS Timer option allows you to configure a QoS session to take effect during a predefined time period by setting the start and end times for the specific control.
- **IEEE 802.1Q priority tag** — When you define a control for a classifier, you can select an IEEE 802.1p priority tag value to insert into forwarded frames. Verify that this priority tag is applied to ports that are configured for IEEE 802.1Q tagging.

- **QoS bandwidth** — Specifies the weighting of the high-priority and best-effort transmit queues. The bandwidth for the control queue is set by means of RSVP. By default, 75 percent of the bandwidth is used for high-priority traffic and 25 percent is used for best-effort packets. (That is, three high-priority packets are sent for each best effort packet). Low-priority packets have no bandwidth allocated.

- **QoS excess tagging** — Allows you to select an IEEE 802.1p priority tag value for nonconforming excess packets (packets that exceed the rate limit). This option refers to any packets marked as excess that you want to tag. After you enable this option, select an IEEE 802.1p priority tag value in the range of from *0* through *7* (`0` is the default). Specifying `1` makes nonconforming excess packets background traffic.

  For nonflow classifiers only, IEEE 802.1P tag values range from 0 through 7. To allow low priority queues to get serviced and to prevent starvation of best effort traffic in the low priority queue, 3Com has implemented the following map:

  - priorities 1-2 map to the low queue
  - priorities 0, 3 map to the best queue
  - priorities 4-7 map to the high queue

  These are the defaults which you can change through modifying the associated classifiers and controls. See classifiers 401 through 407 and 499 and associated controls 1 through 4, using the QoS CLASSIFIER SUMMARY and QoS CONTROL SUMMARY commands respectively.

Key Guidelines for Implementation

Consider the following guidelines when you configure QoS on your Layer 3 switching module.

Procedural Guidelines

Configure classifiers and controls in the following order:

1. Define a classifier, or choose a predefined classifier. Identify a particular type of traffic that you want to regulate and define a classifier for this traffic via the Administration Console. The rules for defining classifiers are different for flow versus nonflow classifiers.
2. Create controls to apply to your classifiers. A control enables the Multilayer Switching Module to direct the traffic to one of the available transmit queues or drop the traffic. When you define a control, you can:
   - **a** Assign a rate limit to the incoming classified traffic (optionally, on a per-port basis).
   - **b** If you specify a rate limit, define what should be done with the nonconforming excess (that is, traffic that exceeds the rate-limit parameters).
   - **c** Apply an IEEE 802.1p priority tag value to forwarded traffic.

General Guidelines

- You must define a classifier before you can assign a control to it.
- A classifier does not affect traffic scheduling until you configure a control for that classifier.
- Traffic that is not classified and controlled is treated with a transmit priority of best (best effort) using the default classifier (499) and the default control (1). All such packets are conforming packets.
- You cannot remove or modify the default classifier *(499).*
- You cannot remove the default control *(1)*, but you can modify it.
- When you specify a TCP or UDP source and destination port range for a flow classifier, limit the range as much as possible (for example, to a single TCP or UDP port or to a small range of ports). If the classifier applies to a wide range of TCP or UDP ports, you increase the amount of classified traffic on the Multilayer Switching Module and consume valuable QoS resources (cache entries).
- If you have defined a control and you want to remove or modify the associated classifier, you must remove the control before you can remove or modify the classifier.

The following items describe how QoS control aggregate rate limit for flow classifiers works on ports that are in certain groups.

- QoS control aggregate behavior for flow classifiers works only on ports that are in certain groups. The aggregate ports are treated as one port for the rate limit specified.
- That is, if a 50-percent rate limit is specified for applicable 100 MB ports, then the aggregate ports total rate in each group adds up to 50 percent of 100 MB or 50 MB.

The following list describes Multilayer Switching Modules and what ports support QoS aggregate rate limit for flow classifiers:

- The 12-port 10/100BASE-TX Fast Ethernet Multilayer Switching Module (Model Number 3CB9RF12R) allows aggregation on ports 1-3, 4-6, 7-9, and 10-12.
- No ports can be aggregated on the 4-Port Gigabit Ethernet Multilayer Switching Module (GBIC) (Model Number 3CB9RG4). Aggregate works the same as ReceivePort if specified.

QoS Classifiers

You define classifiers to distinguish certain types of traffic from other types of traffic. A classifier directs the Multilayer Switching Module how to identify a certain type of traffic. After you define a classifier, you must apply a control to the classifier.

Important Considerations

Review the following considerations before you configure classifiers:

- You can classify bridged or routed traffic (such as AppleTalk or IPX) based on protocol type, cast type, and IEEE 802.1p priority. For routed IP traffic, you can also classify traffic by IP source addresses, destination addresses, or TCP or UDP sockets.
- Before you define a classifier, determine whether you can use one of the Multilayer Switching Module’s predefined classifiers (classifiers that come with your Multilayer Switching Module). If you decide to define your own classifier, you need to decide which type of classifier to define: *flow* (IP routed traffic only) or *nonflow* (bridged or routed traffic).

- You can define up to 100 *flow* classifiers and up to 16 *nonflow* classifiers. Because the Multilayer Switching Module predefines 16 nonflow classifiers, you must delete one of the existing nonflow classifiers (except the default classifier) before you can add your own nonflow classifiers. See “Modifying and Removing Classifiers and Controls” later in this chapter for information about changing or deleting a classifier.
- When you configure a classifier, the Multilayer Switching Module prompts you for different information based on your choice of defining a flow or nonflow classifier.

Using Predefined Classifiers

Figure 66 shows a QoS classifier summary from the Administration Console with the two predefined flow classifiers (FTP and Telnet) and 16 predefined nonflow classifiers, along with their associated controls. (You can use your configuration tool to display summary and detail information for your classifiers.)

The Multilayer Switching Module provides a default classifier (499), which you cannot remove or modify. To first modify one of the predefined nonflow classifiers with controls, you must remove the control.

In Figure 66, U means unicast, M means multicast, and B means broadcast. Also, the range 0 through 7 implies that a nonflow classifier recognizes all IEEE 802.1p priority tags. (See Table 92.)

Figure 66 Predefined Classifiers and Associated Controls

```
        Classifier Name                              Control Cast Protocol  802.1p
Flow   ┌20         FTP                               none    UM   TCP       --
       └23         Telnet Traffic                    none    UM   TCP       --
       ┌401        Background                        2       UMB  any       1
       │402        Standard                          2       UMB  any       2
       │403        Business Critical                 3       UMB  any       3
       │404        Streaming Multimedia              4       UMB  any       4
       │405        Interactive Multimedia            4       UMB  any       5
       │406        Interactive Voice                 4       UMB  any       6
       │407        Network Control                   4       UMB  any       7
Nonflow│420        TCP/IP                            none    U    TCP/IP    0-7
       │430        IP Unicast                        none    U    IP        0-7
       │440        IP Multicast                      none    M    IP        0-7
       │450        IP Broadcast                      none    B    IP        0-7
       │460        IPX Unicast                       none    U    IPX       0-7
       │470        IPX Multicast/Broadcast           none    MB   IPX       0-7
       │480        Appletalk Unicast                 none    U    Appletalk 0-7
       │490        Appletalk Multicast/Broadcast     none    MB   Appletalk 0-7
       └499        Default                           1       UMB  any       0-7
```

Assigning Flow and Nonflow Classifier Numbers

Each classifier requires a unique number in the range of from 1 through 498. When you define a classifier, the first information you supply is the classifier number. The number you specify dictates which type of classifier you are defining.

Default The default classifier number is *499,* which you cannot remove or modify, because all traffic that passes through the QoS engine and the Multilayer Switching Module needs a default classifier to handle all packets.

- To define a flow classifier (for routed IP packets only), specify a value in the range of from *1* through *399*. This setting allows you to specify IP source or destination addresses (or both) as well as TCP or UDP socket information.
- To define a nonflow classifier (for bridged or routed packets), specify a value in the range of from *400* through *498*. (See the list of predefined nonflow classifiers in Figure 66.) For nonflow classifiers, you cannot classify to the IP address or socket level.

The classifier number indicates precedence. The classifier with the *lowest* number takes precedence if a packet meets the criteria for more than one classifier.

For example, you can use two classifiers as follows:

- You define a flow classifier with classifier number 6, which recognizes all TCP or UDP traffic from IP address 3.3.3.3. The control that you assign to this classifier (control 5) gives this traffic a *low*-priority service level.
- You use the predefined nonflow classifier 420, which recognizes all TCP traffic, and create a control for this classifier to give the TCP traffic a *high*-priority service level. (By default, this classifier has no control.)

With these classifiers in place, if 3.3.3.3 sends TCP traffic, this traffic receives *low* priority, because classifier number 6 is lower than classifier 420 and has a higher precedence. Table 93 shows the basic information for these classifiers.

Table 93 Classifier Number Precedence for Two Classifiers

Classifier	Name	Control	Cast	Protocol	802.1p
6 (user-defined)	from_3.3.3.3	5 (for low priority)	UM	all (TCP, UDP)	–
420 (predefined)	TCP/IP	6 (for high priority)	U	TCP/IP	0–7

Defining Flow Classifiers

You can define up to 100 flow classifiers per Multilayer Switching Module for routed IP traffic. When you define a flow classifier (using a unique classifier number), you can create one or more address/port patterns (filters) for that classifier.

Each address/port pattern counts toward the flow classifier limit. Therefore, if you define a flow classifier with 10 address/port patterns, you can have up to 90 additional flow classifiers.

Because a flow classifier handles IP routed traffic only, it is expected that you have an IP VLAN, an IP routing interface, and IP routing enabled. For information on VLANs, see Chapter 14.

Flow Classifier Information

You supply the following information when you define a flow classifier:

- Classifier number in the range of from 1 through 399 (20 and 23 are predefined)
- Classifier name (a unique name of up to 32 characters long)
- Cast type (unicast, multicast, or both). If you create a classifier to block all IP unicast traffic, the Multilayer Switching Module blocks TCP and UDP unicast traffic only, not ICMP traffic.
- IP protocol type (TCP, UDP, or all)
- Source IP address (in standard dot notation, such as 192.101.10.0)
- Source IP address mask (not a subnet mask; see “Specifying Addresses and Address Masks”)
- Destination IP address
- Destination IP address mask
- Start and end of a TCP or UDP source port range as a number from 0 through 65535
- Start and end of a TCP or UDP destination port range as a number from 0 through 65535
- Whether you want to define another address/port pattern (filter) for this classifier

Specifying Addresses and Address Masks

You can classify traffic using source and destination IP addresses and their associated source and destination IP address masks. For a classifier aimed at filtering traffic to a specific destination from a particular source, for instance, you may define a single address/port pattern that specifies the source address and the destination address. Or, if classified traffic to and from certain locations is going to be controlled at the same service level, you may decide to use two address/port patterns: one pattern that covers IP address A as the source and IP address B as the destination, and a second pattern that covers IP address B as the source and IP address A as the destination.

You specify a source or destination IP address in standard dot notation, such as 192.101.10.0. An address of all zeroes is a wildcard match for any source or destination address. Use 0 as a wildcard in any portion of the address.

For the source or destination IP address mask, you specify how many parts of the IP address you want to match. Place a 255 in each portion of the mask that you want the software to recognize; place a 0 in any portion of the mask that you want the software to ignore.

The following examples show several ways to specify IP addresses and IP address masks:

- An IP address of 192.101.20.254 with a mask of 255.255.255.255 requests an exact match for the host IP address 192.101.20.254.
- An IP address of 192.101.20.0 with a mask of 255.255.255.0 requests a match for any node on the subnet 192.101.20.0.
- An IP address of 192.101.20.40 with a mask of 255.255.0.0 requests a match for any node on the 192.101.0.0 network.
- A destination IP address of 0.0.0.254 (or 192.101.20. 254) with a mask of 0.0.0.255 requests a match on any node that ends in 254.
- A mask of 0.0.0.0 is a wildcard match.

Specifying Ports and Port Ranges

Many common applications are associated with well-known port numbers. For example, FTP (which uses TCP) uses port 20 for the data-transfer connection and port 21 for the control connection; TELNET (which also uses TCP) uses port 23; SNMP (which uses UDP) uses port 161; SMTP (the mail protocol) uses port 25; and the World Wide Web service uses port 80. You can consult the services database file (/etc/services on a UNIX server) that is typically associated with TCP/IP hosts for a list of the well-known applications (services) and port numbers. For other applications, you may have to determine the appropriate port number. See RFC 1700 for a list of port assignments for known services.

When you specify the start and end range of a TCP or UDP port, specify as small as range as possible, such as 1 port (for example, port 2049 as both the start and the end of the range). If the classifier applies to a wide range of TCP or UDP ports, you increase the amount of classified traffic on the Multilayer Switching Module and consume valuable QoS resources.

To define flow classifiers and their associated controls for specific scenarios, see “Examples of Classifiers and Controls” later in this chapter.

Defining NonFlow Classifiers

Nonflow classifiers enable you to classify bridged or routed frames according to protocol, cast type, and IEEE 802.1p priority tag values. You can define up to 16 nonflow classifiers per Multilayer Switching Module. The module predefines 16 nonflow classifiers for you. Therefore, to define your own nonflow classifier, you must first delete one of the predefined nonflow classifiers.

NonFlow Classifier Information

You supply the following information when you define a nonflow classifier:

- A classifier number in the range of from 400 through 498. (401 through 407, 420, 430, 440, 450, 460, 470, 480, and 490 are predefined.)
- A classifier name (a unique name of up to 32 characters long)
- A cast type (unicast, multicast, broadcast, or all)
- A protocol type (TCP/IP, IP, IPX, AppleTalk, any or custom)

  If you choose `custom,` select the protocol type (ethernet or DSAP/SSAP)

  - For ethernet type enter the hexadecimal value
  - For DSAP/SSAP type enter the DSAP and SSAP hexadecimal values
- An IEEE 802.1p tag value in the range of from 0 through 7 or all. You can make the Multilayer Switching Module recognize any IEEE 802.1p tagged frames with any combination of the priority tags in the range of from 0 through 7. The tag value is automatically used by the associated control when it forwards frames. See “IEEE 802.1p” earlier in this chapter for more information on the tag values.

For example, you may create a nonflow classifier for your bridged AppleTalk traffic, assign it a cast type of broadcast, a protocol type of AppleTalk, and an IEEE 802.1p tag value of all. You can then apply a control to this classifier to assign a rate limit, service level, and IEEE 802.1p tag to apply to forwarded frames.

For examples of how to define nonflow classifiers and their associated controls for specific scenarios, see “Examples of Classifiers and Controls” later in this chapter.

QoS Controls

After you define a classifier, you assign it a control to apply any of the following values:

- Rate limit (to limit the amount of input bandwidth the classifier uses)
- Service level for conforming packets (a transmit priority that maps to a particular transmit queue)
- Whether packets conforming to the rate limit are loss eligible (that is, whether they are discarded instead of queued when transmit queues back up beyond a threshold)
- IEEE 802.1p priority tag value to apply to forwarded frames
- A one-way filter to drop packets used to establish TCP connections
- Whether to drop packets used to establish TCP connections. This is a form of one-way filtering for flow classifiers only. The default is no.
- Enable control start and stop times. Similar to how a VCR operates, this timer allows you to set the desired beginning and ending period for a control. The default is no.

  If you select yes, you set the following:

  - Input time type such as daily, weekdays, or weekends. You can also choose a specific type that lets you choose an exact day.
  - The starting and ending time expressed in hour, minute, and am or pm (hh:mm), as applicable.
  - The starting and ending date expressed in month and day (MM-DD), as applicable.
- One or more classifiers (classifier numbers) that are subject to this control

Important Considerations

Review the following considerations before you configure controls:

- The Multilayer Switching Module predefines controls 1 through 4 for some of the predefined nonflow classifiers. You can also modify one of these predefined controls. You cannot remove the default control 1, but you can modify it. Also you can assign default control 1 only to classifier 499.
- You can create controls for classifiers in several ways:
  - Apply one control to only one classifier.
  - Apply one control to multiple classifiers.
  - Assign a rate limit of none to a control and thereby emphasize the service level and priority tag.
  - Assign a rate limit type of receivePort or aggregate to the control and define multiple rate-limit values for different subsets of ports.
- Each classifier can have only *one* control. Therefore, although you can apply a control to a classifier that has multiple rate-limit values for subsets of ports, that control can have only one priority specification for forwarded frames. To use different priority levels, use multiple classifiers.

For examples of how controls can be applied to classifiers, see "Examples of Classifiers and Controls" later in this chapter. For information about modifying or removing controls, see "Modifying and Removing Classifiers and Controls" later in this chapter.

Assigning Control Numbers

Each control must have a unique control number. When you define a control, the Multilayer Switching Module provides the next-available control number, but you can specify any unreserved control number.

The Multilayer Switching Module supports control numbers in the range of from 0 through 50 and predefines controls 1 through 4 for some of the predefined nonflow classifiers. Control 1 is associated with the default classifier and can be modified but not removed. You can modify the other predefined controls as well (2 through 4). For example, to redefine the way Business Critical traffic is handled, you may want to modify predefined control 3.

Table 94 lists the predefined controls.

Table 94 Predefined Controls

Control Number/Name	Service Level	Classifiers Controlled	Other Characteristics
1 Default/Best Effort	best	499 (default)	No rate limit, not loss eligible, no priority
2 Background	low	401, 402	No rate limit, not loss eligible, no priority
3 Business Critical	best	403	No rate limit, not loss eligible, no priority
4 Controlled Load	high	404, 405, 406, 407	No rate limit, not loss eligible, no priority

Use your configuration tool (such as the Administration Console) to display summary and detail information for your controls.

When you define a control, you supply the following information:

- Control number in the range of from 5 through 50 (unless you remove the predefined controls from predefined classifiers)
- Control name (a unique name of up to 32 characters long)
- Rate-limit type for the control (none, receivePort, or aggregate)
- Service level (transmit priority) for conforming packets.
- Whether the conforming packets are loss eligible. The default is no.

- For the rate limit type receivePort or aggregate:
  - Service level for nonconforming excess (packets exceeding the rate limit)
  - Whether nonconforming excess are loss eligible
  - How the rate limit is expressed (percentage of port bandwidth or KBps)
  - Rate-limit value (0 through 65434 Kbps or 0 through 100 percent)
  - Burst size in Kbytes (*16* through *8192*; the default value depends on your specified rate limit)
  - Bridge ports for which you want to enable the specified rate limit value (specified bridge ports or all bridge ports). If you specify a subset of available ports, you can enter another rate-limit value for another set of ports.
- For any type of rate limit (and a service level other than drop), any combination of IEEE 802.1p priority tag values in the range of from *0* through *7* or *none* to apply to forwarded frames. By default, no tags are applied unless the associated classifier defines a tag value. In that case, the tag value from the associated classifier is used for the forwarded frames.
- Whether to drop packets used to establish TCP connections. This is a form of one-way filtering for flow classifiers only. The default is no.
- Enable control start and stop times. Similar to how a VCR operates, this timer allows you to set the desired beginning and ending period for a control. The default is no.

  If you select yes, you set the following:

  - Input time type such as daily, weekdays, or weekends. You can also choose a "specific" type that lets you choose an exact day.
  - The starting time expressed in hours, minutes, and am or pm.
  - The ending time expressed in hours, minutes, and am or pm.
  - One or more classifiers (classifier numbers) that are subject to this control.

Specifying Rate Limits

A rate limit restricts the amount of input bandwidth that is used by incoming classified traffic (optionally, on a per-port basis). When you define a control, you can specify one of three rate limits:

- **None** — No rate limit
- **ReceivePort** — Imposes a separate limit on each receive port
- **Aggregate** — Imposes limits on groups of receive ports. This rate limit type can only be applied to flow classifiers.

Your choice of rate limit determines how much additional information you need to supply. The default rate limit is `none`, which means that there is no rate limit applied to the classifier. If you specify a rate limit of `none`, you have a small subset of options to specify. You select a service level and loss-eligibility status for conforming packets (packets that are below the rate limit), decide if you want to apply an IEEE 802.1p priority tag value to forwarded frames (for service levels other than drop), and specify the classifiers with which you want to associate the control.

If you specify a rate limit of `receivePort` or `aggregate`, you have many additional options. After you specify a service level and loss-eligibility status for conforming packets, you can also specify a service level for nonconforming excess packets (packets that exceed the specified rate limit), whether the nonconforming excess are loss eligible, how the rate limit for receive ports should be expressed, the rate-limit value, a burst size, and the receive ports for which you want to enable the rate limit. (The rate limit sets a bandwidth limit for a specific set of ports. You can specify multiple rate-limit values for different subsets of ports. As with any rate limit type, you can additionally specify an IEEE 802.1p priority tag value on forwarded frames.)

When you specify how a receivePort or aggregate rate limit is expressed, you can select a percentage of port bandwidth or KBps:

- For KBps as a rate limit (the default), specify the value for the rate limit in KBps (*0* through *65434*).
- For a percentage for the rate limit, specify the percentage in the range of from *0* through *100* percent. These numbers are rounded to the nearest 16 KB. A value of *0* makes all packets nonconforming excess packets. The Multilayer Switching Module drops these packets only if the service level for excess packets is set to `drop`.

After you specify how the rate limit is expressed, you can specify a burst size. The *burst size* is the maximum amount of data that you can transmit at the line rate before the transmission is policed. This value accommodates variations in speeds and allows you to occasionally exceed the configured rate.

Specifying Service Levels

When you define a control, you specify a service level (a transmit priority). Most of the service levels that you can specify represent a specific transmit queue. You can assign service levels to conforming packets (packets that are within the rate limit) and to nonconforming excess packets (packets that exceed the rate limit).

For information on assigning an IEEE 802.1p priority to nonconforming excess packets, see "QoS Excess Tagging" later in this chapter. For information on the transmit queues and QoS bandwidth, see "Transmit Queues and QoS Bandwidth" later in this chapter.

Service levels also define the loss-eligibility status for conforming and nonconforming excess. By default, conforming packets are *not* loss-eligible; nonconforming excess *are* loss-eligible.

The Multilayer Switching Module supports these service levels:

- **High** — For any type of rate limit, transmits the packet first (top priority)
- **Best** — For any type of rate limit, transmits the packet on a best-effort basis (the default for conforming and nonconforming excess packets)
- **Low** — For any type of rate limit, transmits the packet on a low-priority basis
- **Drop** — For a rate limit of none, drops *all* conforming packets on *all* ports associated with the classifiers. For a rate limit of receivePort or aggregate, drops all nonconforming excess packets.

  If you want to drop conforming packets for only a subset of ports, use the receivePort or aggregate rate limit, set the rate limit to 0, and specify the group of ports.

  If you specify drop for the service level for conforming packets (that is, you are using a rate limit of none), the Multilayer Switching Module does not give you the option of specifying an IEEE 802.1p tag.

Specifying TCP Drop Control

The TCP drop control option lets you create a control for packets used to establish TCP connections. This control affects QoS Flow Classifiers that have TCP traffic going from *source* IP addresses to *destination* IP addresses.

TCP drop control does not function with nonflow classifiers or UDP. It is only available for flow classifiers that include TCP.

Figure 67 illustrates how TCP handshaking works between the source and destination to establish a connection. By dropping only the *initial* TCP packet used to establish TCP connections (those packets containing a signature of SYN=1, ACK=0), you can establish one-way TCP flow filtering.

Figure 67 TCP Handshaking

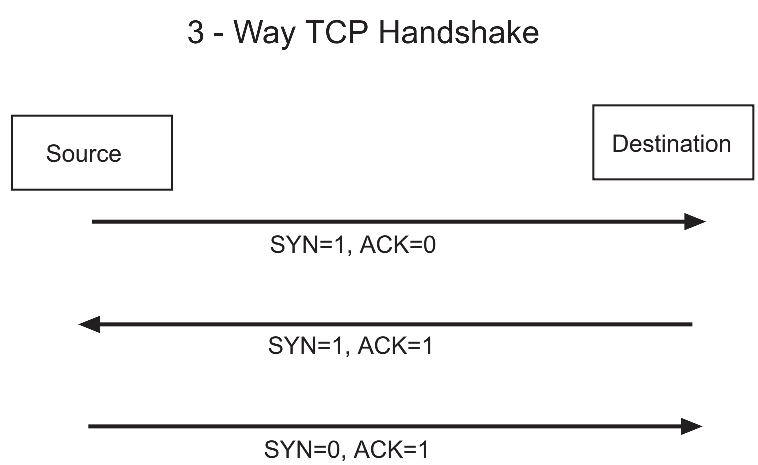
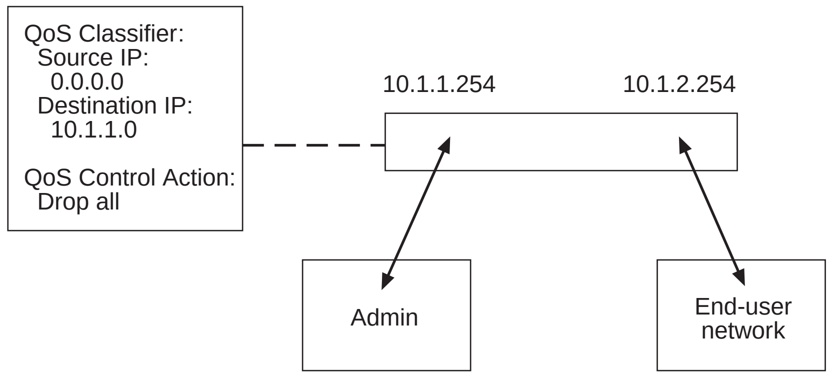


Figure 68 shows an example with TCP drop control disabled.

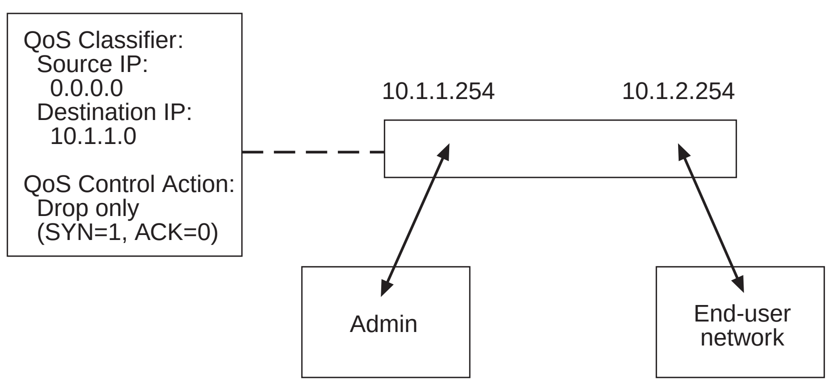
Figure 68 QoS Control Action (Drop Control Disabled)



With the QoS Classifier and QoS Control definition shown in Figure 68 (TCP control is not enabled), any attempt by a client on the End-user network to establish a TCP connection to a server on the Admin network fails.

This next example illustrates how TCP one-way-filtering can be effective. Figure 69 shows the same situation, but with TCP drop control enabled to filter only those packets with the SYN=1 and ACK=0 signature.

Figure 69 QoS Control Action (Drop Control Enabled)



In this example, any attempt by a client on an End User network to establish a TCP connection to a server on the Admin network still fails, but it is now possible for clients on the Admin network to establish TCP connections to servers on any network without restriction.

Setting the QoS Timer Control

The QoS Timer option allows you to configure a QoS session to take effect during a predefined time period by setting the start and end times for the specific control. Setting the start and end times is similar to using a VCR to record programs.

The default setting for the timer control is no (no timer control). QoS controlled classifiers are in effect all the time when timer control is not enabled.

- Starting and ending days in the following syntax: `mm-dd`
  For example, to enter a date of May 20, enter **`05-20`**
- Starting and ending times in the following syntax: `hh:mm`
  For example, to enter a time of 10 o'clock in the morning, enter **`10:00`**
- Days of the week
  (Monday=1, Tuesday=2, Wednesday=3, Thursday=4, Friday=5, Saturday=6, Sunday=7). For example, to enter Monday as the day of the week, enter **`1`**
- You can verify the timer control options using the `qos control detail` command. The detail displays the type of time control, the start and end times, and the classifiers associated with the control.

The time is verified every minute.

Timer Options

The following options are available for the timer control:

- **Specific Day** — Select the specific start day and time, and the specific end day and time. The control is removed after the end time is reached.
- **Daily** — Select a starting day and then a start and end time. The control is activated between the start and end time everyday.
- **Day of the Week** — Select a day and then a start and end time. The control is removed after the end time is reached.
- **Every Day of the Week** — Select a start day and then the start time and end time. The control is activated between the start and end times every 7 days.
- **Weekdays** — Select start and end time. The control is activated every weekday between the start and end times for the current week.
- **Weekends** — Select a start and end time. The control is activated during each day of the current weekend and is removed when the Sunday end time is reached.
- **Every Week Day** — Select a start and end time. The control is activated between the start and end times every weekday.
- **Every Weekend** — Select a start and end time. The control is activated on each weekend day between the start and end times.

Examples of Classifiers and Controls

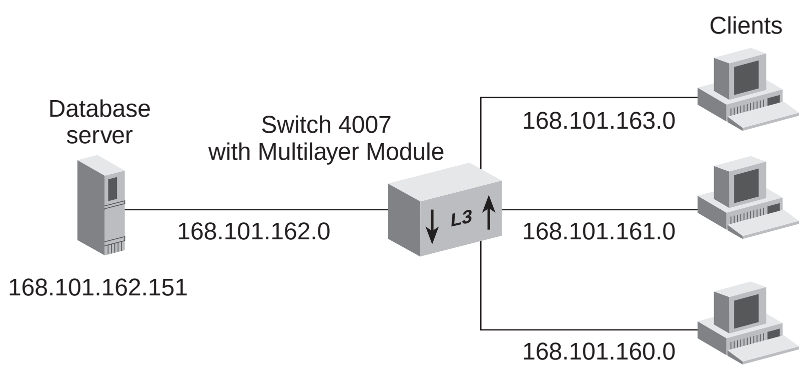
The following examples show ways to implement flow and nonflow classifiers and their associated controls.

In all examples, a Multilayer Switching Module on the Switch 4007 provides the illustrated connections.

Example 1: Traffic To/From a Specific Server

In the first example, a flow classifier is defined with two address and port patterns (filters) to classify traffic from subnetworks of the 168.101.0.0 network *to* the database server 168.101.162.151, and traffic *from* the server to the subnetworks. This kind of configuration can be called a to/from classifier. The control applied to this classifier gives high priority to the traffic to and from the server.

Figure 70 To/From Flow Classifier and Control for Server Traffic



To/from classifier definition with two address and port patterns:

Classifier Field	Classifier Definition
Classifier number	15
Classifier name	DBServer1
Cast type	unicast
IP protocol type	UDP
Source IP address	168.101.0.0
Source IP address mask	255.255.0.0
Destination IP address	168.101.162.151
Destination IP address mask	255.255.255.255
Start/end source port range	2020/2020
Add another filter (address/port pattern)?	y
Source IP address	168.101.162.151
Source IP address mask	255.255.255.255
Destination IP address	168.101.0.0
Destination IP address mask	255.255.0.0
Start/end destination port range	2020/2020
Add another filter (address/port pattern)?	n

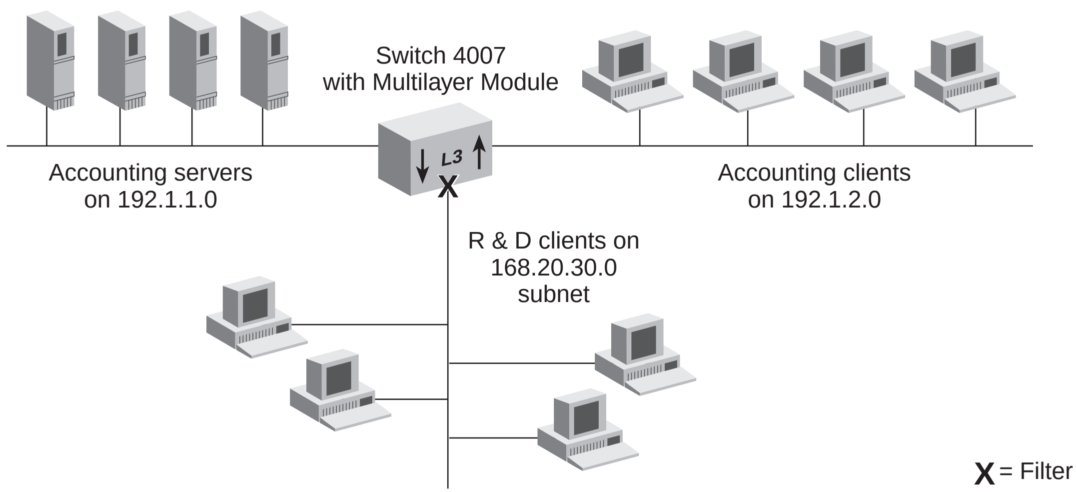
The control definition for the to/from classifier:

Control Field	Definition
Control number	5
Control name	*DBServer1*
Rate limit type	*none*
Service level	high
Loss eligible status	no
802.1p tag for forwarded frames	none
Classifiers controlled	15

Example 2: Filtering Traffic to a Destination

In the following example, a flow classifier is defined to block access to the Accounting network 192.1.0.0 (which includes subnetworks 192.1.1.0 and 192.1.2.0) from the Research and Development 168.20.30.0 subnetwork. The associated control for this classifier sets a service level of *drop* to drop all traffic that is sent by the 168.20.30.0 subnet to the Accounting network.

Figure 71 Flow Classifier for Traffic to/from a Subnetwork



Classifier definition for filtering traffic to a specific destination:

Classifier Field	Classifier Definition
Classifier number	26
Classifier name	*IPFilter1*
Cast type	*all*
IP protocol type	all
Source IP address	168.20.30.0
Source IP address mask	255.255.255.0
Destination IP address	192.1.0.0
Destination IP address mask	255.255.0.0
Start/end source/destination port range	0/65535
Add another address/port pattern?	n

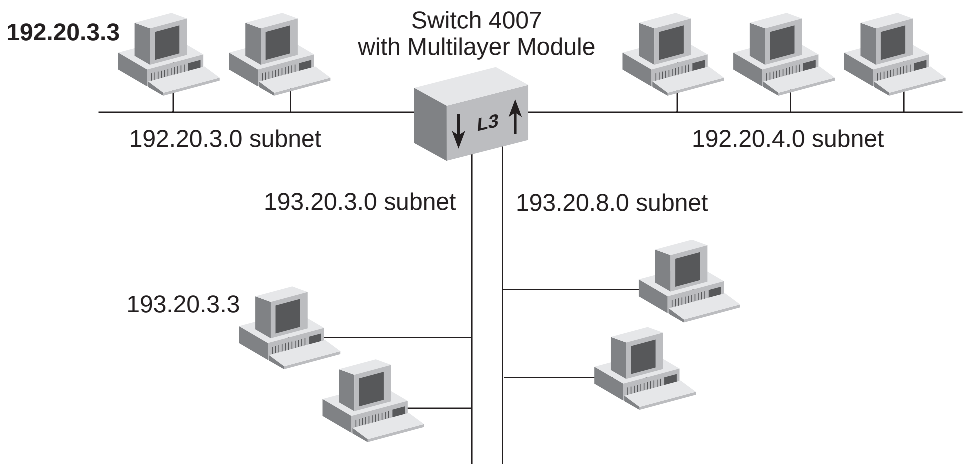
The control definition for this filtering classifier:

Control Field	Definition
Control number	6
Control name	IPFilter1
Rate limit type	none
Service Level	drop
Classifiers controlled	26

Example 3: Using Two Classifiers to Filter Traffic

In the following example, two flow classifiers (1 and 3) are defined with controls to filter IP traffic. Classifier 1 permits IP traffic between two hosts (192.20.3.3. and 193.20.3.3), while classifier 3 drops IP traffic TCP and UDP, not ICMP) to and from one of the hosts (192.20.3.3). This example shows how the classifier number can be used to dictate precedence.

Figure 72 Flow Classifier for Traffic to/from a Subnetwork



First classifier definition for filtering traffic to/from a specific destination:

Classifier Field	Classifier Definition
Classifier number	1
Classifier name	192.20.3.3_to_193.20.3.3
Cast type	all
IP protocol type	all
Source IP address	192.20.3.3
Source IP address mask	255.255.255.255
Destination IP address	193.20.3.3
Destination IP address mask	255.255.255.255
Start/end source port range	0/65535
Add another filter (address/port pattern)?	y
Source IP address	193.20.3.3
Source IP address mask	255.255.255.255
Destination IP address	192.20.3.3
Destination IP address mask	255.255.255.255
Start/end destination port range	0/65535
Add another filter (address/port pattern)?	n

The control definition for the first filtering classifier:

Control Field	Definition
Control number	5
Control name	192.20.3.3_to_193.20.3.3
Rate limit type	none
Service level	best
802.1p tag for forwarded frames	none
Classifiers controlled	1

Second classifier definition for filtering traffic to/from a specific destination:

Classifier Field	Classifier Definition
Classifier number	3
Classifier name	192.20.3.3_to_all
Cast type	all
IP protocol type	all
Source IP address	192.20.3.3
Source IP address mask	255.255.255.255
Destination IP address	0.0.0.0 (all)
Destination IP address mask	0.0.0.0
Start/end source port range	0/65535
Add another filter (address/port pattern)?	y
Source IP address	0.0.0.0 (all)
Source IP address mask	0.0.0.0
Destination IP address	192.20.3.3
Destination IP address mask	255.255.255.255
Start/end destination port range	0/65535
Add another filter (address/port pattern)?	n

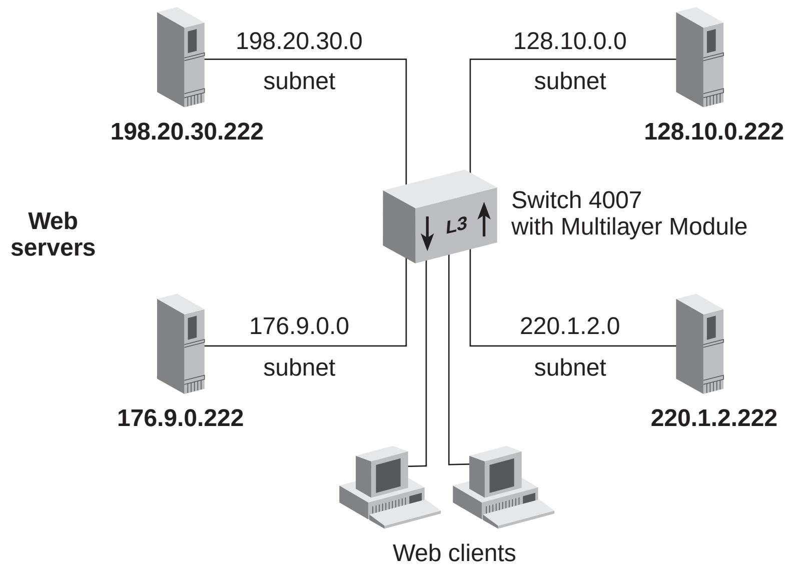
The control definition for the second filtering classifier:

Control Field	Definition
Control number	7
Control name	192_20.3.3_to_all
Rate limit type	none
Service level	drop
Classifiers controlled	3

Example 4: Assigning High Priority to Specific Traffic

In the following example, a classifier is defined to give high priority to Web server (http) traffic. In this configuration, all Web servers have addresses that end in .222. This example can apply to any type of traffic that needs high priority (for example, mail server traffic).

Figure 73 Flow Classifier for Assigning High Priority to Web Traffic



Classifier definition for high-priority Web traffic:

Classifier Field	Classifier Definition
Classifier number	17
Classifier name	httpServer1
Cast type	unicast
IP protocol type	TCP
Source IP address	0.0.0.0
Source IP address mask	0.0.0.0
Destination IP address	0.0.0.222
Destination IP address mask	0.0.0.255
Start/end source port range	80/80
Add another filter (address/port pattern)?	y
Source IP address	0.0.0.222

Classifier Field	Classifier Definition
Source IP address mask	0.0.0.255
Destination IP address	0.0.0.0
Destination IP address mask	0.0.0.0
Start/end destination port range	80/80
Add another filter (address/port pattern)?	n

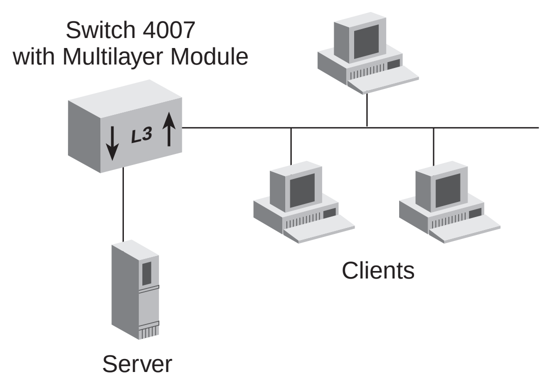
The control definition for this classifier is as follows:

Control Field	Definition
Control number	7
Control name	httpServer1
Rate limit type	none
Service level	high
802.1p tag for forwarded frames	none
Classifiers controlled	17

Example 5: Nonflow Multimedia Tagged Traffic

In this example, a nonflow classifier is defined to classify bridged multimedia traffic with an IEEE 802.1p priority tag of 5 and control this traffic with a high priority transmit service level and a rate limit of 2048 Kbps.

Figure 74 Nonflow Classifier/Control for Bridged Multimedia Traffic



Nonflow classifier definition for Multimedia Traffic with priority tagging:

Classifier Field	Classifier Definition
Classifier number	405
Classifier name	Interactive Multimedia
Cast type	all (unicast, multicast broadcast, UMB)
Protocol type	any
IEEE 802.1Q tag(s)	5

The control definition for this classifier is as follows:

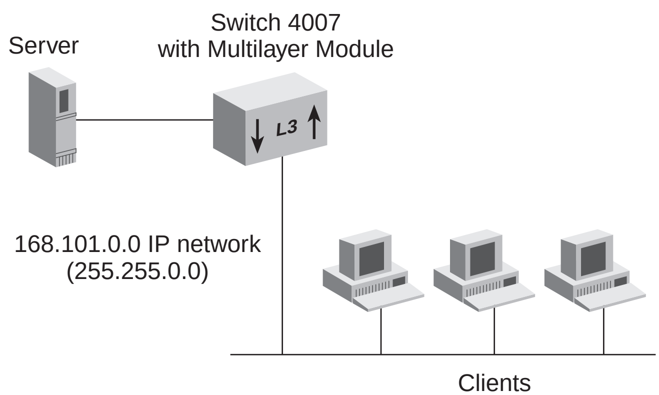
Control Field	Definition
Control number	4
Control name	Interactive_Multimedia
Rate limit type	receivePort
Service level	high
Loss eligible status	no
Excess service level	drop
Excess loss eligible status	–
Representation of rate limit	Kbytes/sec
Rate limit value	2048 KB
Burst size	181 KB
Bridge ports	1 through 13
802.1p tag for forwarded frames	– (uses tag from classifier: 5)
Classifiers controlled	405

Example 6: Bridged Nonflow IP Unicast Traffic

In this example, a nonflow classifier is defined to classify IP unicast traffic between clients and the server on the 168.101.0.0 network.

The applied control handles this *bridged* traffic with a high-priority transmit service level and a rate limit of 75 percent of the link bandwidth.

Figure 75 Nonflow Classifier/Control for Bridged IP Unicast Traffic



Nonflow classifier definition for bridged IP unicast traffic:

Classifier Field	Classifier Definition
Classifier number	430
Classifier name	IP_Unicast
Cast type	unicast (U)
Protocol type	IP
IEEE 802.1Q tag(s)	0 through 7

The control definition for this classifier is as follows:

Control Field	Definition
Control number	5
Control name	IP_Unicast
Rate limit type	receivePort
Service level	high
Loss eligible status	no
Excess service level	low
Excess loss eligible status	yes
Representation of rate limit	percent
Rate limit value	75 percent
Burst size	363 KB
Bridge ports	1 through 13
802.1p tag for forwarded frames	– (uses tags from classifier: 0 through 7)
Classifiers controlled	430

Modifying and Removing Classifiers and Controls

You can modify or remove a previously defined classifier or control. When you modify or remove a classifier, you specify the classifier number; when you modify or remove a control, you specify the control number.

You may want to modify a classifier to alter source/destination information (flow classifier) or change IEEE 802.1p values (nonflow classifier). You may want to modify a control to specify a different service level (queue) or rate limit.

Important Considerations

Before you modify or remove classifiers or controls, review these guidelines:

- You cannot remove the default classifier or the default control, but you can modify the default control. You can modify other predefined classifiers and the predefined controls (for example, if you want to redefine the handling of Business Critical traffic, which is associated with predefined control 3).
- When you apply a control to a classifier, you must remove the control for a classifier before you can modify or remove the classifier.
- When you remove a control, the associated classifiers are no longer controlled and no longer have a rate limit, service level, or 802.1p tag.
- If you want to modify a classifier that has several address/port definitions, you must supply them again during the modification process. If you do not reenter them, the Multilayer Switching Module deletes these definitions.
- If you want to modify a control that uses the rate-limit type of aggregate or receivePort with several rate-limit values, you can change one rate-limit value without affecting the other defined rate-limit values.

QoS Excess Tagging

Your Multilayer Switching Module enables you to tag nonconforming excess packets (that is, packets that exceed the rate-limit criteria) with a special IEEE 802.1p tag value. This tag refers to any packets marked as excess that you want to tag. By default, excess tagging is disabled.

You can use your configuration tool (for instance, the Administration Console) to enable or disable excess tagging and display your excess tagging information.

If you *enable excess tagging*, you can specify an IEEE 802.1p tag value for nonconforming excess packets in the range of from *0* through *7*, with *0* as the default. (See “IEEE 802.1p” earlier in this chapter for a list of the tags and their associated priority levels). Specifying *1* means that nonconforming excess packets become background traffic.

Example: QoS Excess Tagging

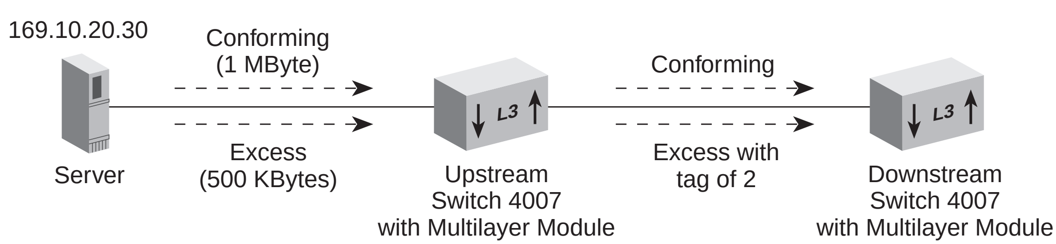
The following example shows how to use a classifier, control, and QoS excess tagging to tag conforming QoS multicast video traffic from a server as *Streaming Multimedia* 802.1p service and to tag any excess traffic as *Standard* 802.1p service.

In this sample configuration:

- The configured rate limit is 1 MB, so when the server sends 1.5 MB, the upstream system knows that 1 MB is conforming and 500 KB is excess.
- The upstream system configures the classifier, control, and the tagging, and has the QoS flow. The upstream system passes the excess traffic with the tag of 2 (standard priority) to the downstream system.
- The downstream system can prioritize traffic from this flow at Layer 2, using its default 802.1p classifier 404 (for conforming packets) and classifier 402 (for nonconforming excess packets) along with the corresponding controls 4 and 2.

For this configuration, you must enable QoS excess tagging with a tag value of 2 as well as define the classifier and control.

Figure 76 QoS Excess Tagging



Classifier definition for QoS Excess Tagging:

Classifier Field	Classifier Definition
Classifier number	25
Classifier name	VideoServer1
Cast type	multicast
IP protocol type	UDP
Source IP address	169.10.20.30
Source IP address mask	255.255.255.255
Destination IP address	0.0.0.0
Destination IP address mask	0.0.0.0
Start/end source/destination port range	2010/2020
Add another filter (address/port pattern)?	n

The accompanying control definition:

Control Field	Definition
Control number	5
Control name	VideoServer1
Rate limit type	receivePort
Service level	high
Loss eligible status	no
Excess service level	low
Excess loss eligible status	yes
Representation of rate limit	Kbytes/sec
Rate limit value	1024
Burst size	128
Bridge ports	all (1 through 19)
802.1p tag for forwarded frames	4
Classifiers controlled	25

Transmit Queues and QoS Bandwidth

QoS uses four transmit queues:

- **Control queue** — The transmit queue for reserved network control traffic, such as RIP or OSPF updates, as well as RSVP data flows. This queue is always serviced first. Bandwidth for this queue is set via RSVP.
- **High priority queue** — The transmit queue with the second highest priority. You can map classifiers directly to this queue.
- **Best effort queue** — The transmit queue used by default for all traffic except reserved traffic.
- **Low priority queue** — The transmit queue with the lowest priority. All traffic assigned to this queue is forwarded only if there is bandwidth still available after the other queues are serviced. Low priority packets do not have bandwidth allocated.

You can configure the weighting of the high-priority and best-effort transmit queues by using the option to modify QoS bandwidth. By default, the weighting of the queues is 75 percent high-priority traffic and 25 percent best-effort traffic. Keep in mind that the weighting does not represent guaranteed output bandwidth for these queues, because they are served in relative percentages after the control queue is serviced.

When you modify the QoS bandwidth, you specify the percentage of bandwidth to be used for the high-priority transmit queue on the output link. You can specify a value in the range from *0* through *100*. The value that you specify determines the ratio of high-priority to best-effort traffic, as follows:

- The value *75* (the default) specifies that three high-priority packets are transmitted for each best effort packet (ratio of 75/25).
- The value *50* sets equal priority for high-priority and best-effort packets (ratio of 50/50).
- The value *100* is strict prioritization; it allows best effort packets to be sent only when no high-priority packets need to be sent.

No bandwidth is ever lost. Because QoS uses ratios, any unused bandwidth can be used by a lower-priority queue.

RSVP

The Resource Reservation Protocol (RSVP) is an IP service that prevents real-time traffic such as voice or video from overwhelming bandwidth resources. In general, RSVP supports QoS IP flow specifications by placing and managing resource reservations across the network (setting admission control, policing, and restricting the creation of RSVP reservations). Your Multilayer Switching Module can reserve and police the bandwidth requested for each RSVP session.

RSVP is receiver-oriented, that is, an end system can send an RSVP request on behalf of an application to request a specific QoS from the network. At each hop along the path back to the source, routers such as your Layer 3 switching module register the reservation and try to provide the required QoS. If a router cannot provide the required QoS, its RSVP process sends an error to the end system that initiated the request.

RSVP is designed for multicast applications, but it also supports resource reservations for unicast applications as well as point-to-point transmissions. RSVP does not implement a routing algorithm.

To use RSVP, you must be routing. (RSVP operates at Layer 3 for IP-based data flows.) End stations in the configuration must support RSVP in order to request the reservation of bandwidth through the network.

By default, RSVP is disabled on the Multilayer Switching Module. If you decide to use RSVP, 3Com recommends that you use the default RSVP settings.

RSVP Terminology

Familiarize yourself with the following RSVP terms:

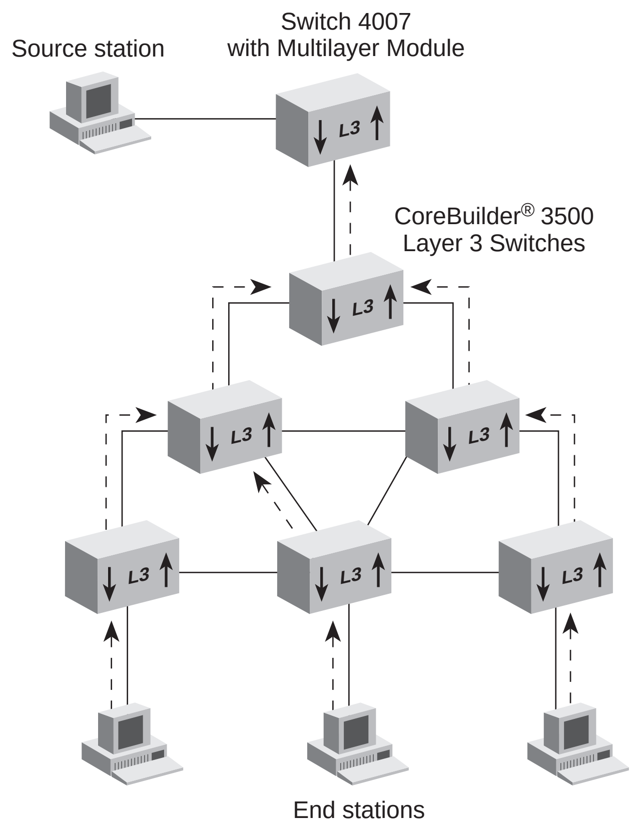
- **RSVP Flow** — A data stream that operates in simplex, going one way from the origin to multiple destinations. The flows go from a set of senders to a set of receivers.
- **Reservation Style** — The types of multicast flows that RSVP installs:
  - **Fixed Filter (distinct) Style** — A flow that originates from one sender only (for example, a video application). This style requires a separate reservation per sender on each transmission type.
  - **Shared Explicit** — A shared-reservation flow that originates from a limited number of senders (for example, an audio application). This style identifies the flows for specific network resources. A single reservation can be applied to all senders in the set.
  - **Wildcard Filter** — A shared-reservation flow from all senders.
- **Total reservable bandwidth percentage** — Controls the admission control policy. RSVP begins to refuse reservations when the requested bandwidth on an output link exceeds the total reservable bandwidth. You specify a percentage of the output link (a value of from *0* through *200,* with *50* as the default). This percentage is the amount of bandwidth that you allow RSVP to reserve in the Multilayer Switching Module. You can over-subscribe (over 100) and specify a value up to 200.
- **Maximum per-reservation bandwidth** — The largest reservation that RSVP attempts to install. Specify this bandwidth using a percentage of the output link (a value of from *0* through *100; 50* is the default).
- **Policing options** — Ensure that an RSVP session uses only as much bandwidth as it requested. The policing options mandate when to drop nonconforming excess packets. You configure the Multilayer Switching Module to observe one of these policing options:
  - **Edge** — Causes nonconforming excess packets to be dropped only at the edge (that is, when the traffic has not yet passed through any network device that has already performed policing for that flow). The Multilayer Switching Module polices the flow when RSVP requests it. Edge is the default policing option. The RSVP protocol knows how to detect what is edge and what is not when it polices.
  - **Always** — The Multilayer Switching Module always polices the flow.

- **Never** — The Multilayer Switching Module never polices the flow, even if RSVP requests it.

Example: RSVP

Figure 77 shows an RSVP configuration in which an RSVP reservation request (dotted lines) flows upstream along a multicast delivery tree (with routing-capable devices such as Switch 4007 Multilayer Switching Modules until it merges with another reservation request for the same source.

Figure 77 Sample RSVP Configuration



Setting RSVP Parameters

If you enable RSVP, you specify the following information:

- Maximum total reservable bandwidth
- Maximum per-reservation bandwidth
- Policing option (*edge, always,* or *never,* with *edge* as the default)
- Service level for excess/policed traffic (*best* or *low*, with *low* as the default). This setting applies to the excess traffic with the reserved bandwidth (that is, in which queue it should be placed).

After you enable RSVP, you can use your configuration tool (for example, the Administration Console) to display summary or detail information about RSVP. Figure 78 shows a sample RSVP detail display.

Figure 78 identifies the RSVP data flow as it passes through the Layer 3 module and provides the following information:

- Session information, including destination IP addresses and ports, protocols, sender, receivers, and RSVP reservations
- Session sender information that identifies port numbers, source IP addresses, previous hop addresses, Logical Interface Handle (LIH) values, Time To Die (TTD) values, bandwidth values, burst size values, and output ports
- Session receiver and reservation information that identifies port numbers, an RSVP style (ST) of fixed filter (FF), shared explicit (SE), or wildcard filter (WF), next-hop addresses, LIH values, TTD, values, bandwidth values, burst size values, and filters
- The actual flow that is installed on the system (shown in the last portion of the display)

Figure 78 RSVP Information with Installed Flows

```
Total Resv Per Resv    Policing  Excess   Excess Loss
Bandwidth  Bandwidth   Option    Service  Eligible
50%        50%         always    low      no


Session    Destination IP:Port   Protocol    Senders  Receivers Reservations
1          228.8.8.8:80          UDP               1          1            1
2          230.2.2.2:20          UDP               1          1            1

Session-Sender          Port Source IP:Port                          Out ports
1-1                        4 158.101.232.50:32827                            8
2-2                        8 158.101.238.9:32809                             4

Session-Sender Previous Hop      LIH TTD  Bandwidth      Burst
1-1            158.101.232.50      0 155      42880      16000
2-2            158.101.90.22       1 144      54784      16000

Session-Receiver        Port Next hop        ST Filter IP:Port
1-1                        8 158.101.90.22   FF 158.101.232.50:32827
2-2                        4 158.101.232.50  FF 158.101.238.9:32809

Session-Receiver                  LIH TTD  Bandwidth      Burst
1-1                                 8 148      42880      16000
2-2                                 4 152      54784      16000

Session-Reservation     Port Next hop        ST Filter IP:Port
1-1                        8 158.101.90.22   FF 158.101.232.50:32827
2-2                        4 158.101.232.50  FF 158.101.238.9:32809

Session-Reservation               LIH TTD  Bandwidth      Burst
1-1                                 8 148      42880      16000
2-2                                 4 151      54784      16000

Session-Installed Flows
                   Port Source IP:Port       Destination IP:Port   Pro   Misses
1                     4 158.101.232.50:32827  228.8.8.8:80          UDP        1
2                     8 158.101.238.9:32809   230.2.2.2:20
```

23 DEVICE MONITORING

This chapter provides descriptions and key operational information about device monitoring features and tools available in your Switch 4007 modules.

These features are available on Switch 4007 Layer 2 and Multilayer Switching Modules. Differences in implementation between these module groups are noted where applicable.

The chapter covers these topics:

- Chapter Scope
- Device Monitoring Overview
- Key Concepts and Tools
- Event Logging
- Baselining
- Roving Analysis
- Ping
- traceRoute
- SNMP
- Remote Monitoring (RMON)
- Management Information Base (MIB)

Chapter Scope

Device monitoring features for the Switch 4007 Enterprise Switch are implemented on the Enterprise Management Engine (EME) module, the Layer 2 and Multilayer Switching Modules, and the switch fabric modules.

Features implemented on the EME module include:

- Event Logging
- Ping
- Simple Network Management Protocol (SNMP)

For information about using these features, see the chapters in Part II of this guide.

Features implemented on Switch 4007 modules include:

- Baselining
- Roving Analysis
- Ping (EME and Layer 3 only)
- traceRoute (Layer 3 only)
- SNMP

Features implemented on all switching modules include:

- RMON
- Management Information Bases (MIBs)

You can perform device monitoring in either of these ways:

- *From the EME or Administration Console menus. You can manage features that are implemented in the EME module using the EME command line interface as described in the chapters in Part II of this guide.*
- *From the respective folders of the Web Management software. See the* Switch 4007 Getting Started Guide. *Not all EME or Administration Console functions are implemented in the Web Management software.*
- *You manage features that are implemented in the interface module and switch fabric module from the menus of the Administration Console after you log in to the Enterprise Management Engine and connect to a module in the Switch 4007 chassis. For information about the Administration Console, see the Command Reference Guide.*

RMON MIBs are accessible only through applications that implement SNMP.

The management interfaces display “`cb9000`*” and refer to the Management Module as the Enterprise Management Engine (EME) because the heritage of the Switch 4007 is the CoreBuilder® 9000 switch.*

Device Monitoring Overview

You can use the device monitoring features and tools described in this chapter to analyze your network periodically and to identify potential network problems before they become serious. To identify potential problems in your network, use:

- Event logging
- Baselining
- Roving analysis
- RMON information

To test and validate paths in your network, use:

- Ping
- traceRoute
- polling MIBs via SNMP

The SNMP protocol and the Management Information Base (MIB) are described in this chapter to give you some background on how performance data is collected on the network.

Key Concepts and Tools

Key concepts and tools for the device monitoring of your system are described in this section to give you a perspective of the scope of device monitoring.

Administration Console

The Administration Console provides you with access to all the features of your system. You can use the Administration console after you log in to the EME and connect to a module slot in the Switch 4007 chassis.

Web Management Tools

The Web Management tools provides you access to the EME and administration console remotely via the internet. It provides you with complete access to the Administration Console as if you are connected locally or through a Telnet connection. See the *Switch 4007 Getting Started Guide* for more information.

Network Management Platform

The network management platform allows you to view the health of your overall network. With the platform, you can understand the logical configuration of your network and configure views of your network to understand how devices work together and the role they play in the users' work. The network management platform that supports your Transcend® Network Control Services software installation can provide valuable troubleshooting tools.

SmartAgent Embedded Software

Traditional SNMP management places the burden of collecting network management information on the management station. In this traditional model, software agents collect information about throughput, record errors or packet overflows, and measure performance based on established thresholds. Through a polling process, agents pass this information to a centralized network management station whenever they receive an SNMP query. Management applications then make the data useful and alert the user if there are problems on the device.

For more information about traditional SNMP management, see "SNMP" later in this chapter.

SmartAgent® software, which uses Remote Monitoring (RMON), is self-monitoring, collecting and analyzing its own statistical, analytical, and diagnostic data. In this way, you can conduct network management by exception — that is, you are only notified if a problem occurs. Management by exception is unlike traditional SNMP management, in which the management software collects *all* data from the device through polling.

Event Logging

The event log messages display real-time information about the state of the system, a specific service, or both, and can help you diagnose site-specific problems.

Event Logging is implemented from the EME module. See the chapters in Part II of this guide for more information.

Baselining

Normally, statistics for MACs and ports start to compile when you turn the system on. Baselining allows you to view statistics compiled over the period of time since a baseline was set. By viewing statistics relative to a baseline, you can more easily evaluate recent activity in your system or on your network.

Baselining is implemented for both Layer 2 and Multilayer Switching Modules of the Switch 4007 system.

Important Considerations

- Baselining is maintained across Administration Console sessions. Statistics that you view after setting the baseline indicate that they are relative to the baseline. To view statistics as they relate only to the most recent power-on, disable the baseline.
- Baselining affects the statistics that are displayed for Ethernet ports and bridges.

Displaying the Current Baseline

You can get a display the current baseline to see when the baseline was last set and to determine if you need a newer baseline for viewing statistics.

Setting a Baseline

You can reset the baseline counters to zero (*0*). The system maintains the accumulated totals since power-on. The baseline is time-stamped.

Enabling or Disabling Baselines

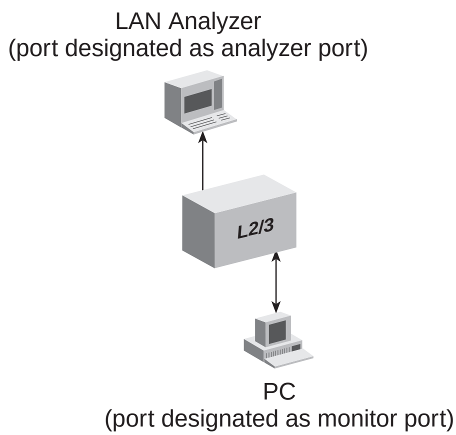
When you reenable a baseline, the counters return to the values that accumulated since the most recent baseline that you set. Disabling a baseline returns the counters to the total accumulated values since the last power up.

Roving Analysis

Roving analysis is the mirroring of Fast Ethernet, Gigabit Ethernet, or Fiber Distributed Data Interface (FDDI) port traffic to another port of the same media type. This second port has an external RMON-1/RMON-2 probe or analyzer attached such as the 3Com Transcend Enterprise Monitor. Through the probe, you can monitor traffic on any switched segment. Figure 79 shows a sample configuration.

- The port with the analyzer attached is called the *analyzer port*.
- The port that is being monitored is called the *monitor port*.

Figure 79 Connecting an Analyzer to the System



Roving analysis is implemented for Layer 2 and Multilayer Switching Modules of the Switch 4007 system.

The monitor port and the analyzer port must *be on the same module.*

The purpose of roving analysis is to:

- Analyze traffic loads on each segment so that you can continually optimize your network loads by moving network segments
- Troubleshoot switched network problems (for example, to find out why a particular segment has so much traffic)

When you set up a roving analysis configuration, the system copies both transmit and receive port data and forwards it to the port on which the network analyzer is attached — without disrupting the regular processing of the packets.

Key Guidelines for Implementation

To enable the monitoring of ports on a system, follow these general steps:

1. Add the port on which you want to attach the network analyzer.
2. Start roving analysis.
   a. Select the port that you want to monitor.
   b. Enter the analyzer port's MAC address.

The system provides commands to add and remove (define and undefine) the analyzer port, to display the current analyzer and monitor ports, and to start and stop analysis.

See the "Roving Analysis" chapter in the *Command Reference Guide* for details.

Important Considerations

- The monitor port and the analyzer port must be on the same module.
- On Layer 2 modules, you can connect one network analyzer to a module
- On Multilayer Switching Modules, you can connect as many network analyzers as there are ports on the module, up to a maximum of 12.
- The network analyzer cannot be located on the same bridge port as the port that you want to monitor.
- For more accurate analysis, attach the analyzer to a dedicated port instead of through a repeater.
- When the analyzer port is set, it cannot receive or transmit any other data. Instead, it receives only the data from the port(s) to be monitored.
- If Spanning Tree Protocol was enabled on the analyzer port, it is automatically disabled. When the analyzer port is undefined, the port returns to its configured Spanning Tree state and functions as it did before it was set as an analyzer port.
- When you configure a port that is part of a virtual LAN (VLAN) as an analyzer port, the port is removed from all VLANs of which it is a member. When you remove the analyzer port, it becomes a member of the default VLAN. You have to manually add it back to its original VLANs.
- You cannot use roving analysis to monitor trunk ports or resilient link ports.

- If the physical port configuration changes in the system (that is, if you remove or rearrange modules), the MAC address of the analyzer port remains fixed. If the module with the analyzer port is replaced with a different media type module, the RAP configuration is cleared.
- On any Layer 3 module, defining a monitor port (`analyzer start` command) affects that port's ability to collect RMON data. The ability to collect RMON data is not lost when the analyzer port is defined (`analyzer add`), but when the monitor port is defined (`analyzer start`). Table 95 lists which RMON groups can continue to collect data, and which cannot after the port has become a monitor port.

Table 95 Roving Analysis and RMON Data

RMON Groups	Works with Roving Analysis?
RMON-1 Groups	
Statistics	Yes
History	Yes
Alarm	Yes
Hosts	No
HostTopN	No
Matrix	No
Event	Yes
RMON-2 Groups	
protocolDir	Yes
protocolDist	Yes
addressMap	No
nlHost	No
nlMatrix	No
alHost	No
alMatrix	No
probeConfig. probeCapabilities	Yes

The RMON groups that require samples of traffic from the ASICs will not work because they do not receive any traffic data when a port is defined as a monitor port. The Multilayer Switching Modules are capable of doing either roving analysis or traffic sampling, but not both at the same time.

Ping

The ping feature is a useful tool for network testing, performance measurement, and management. It uses the Internet Control Message Protocol (ICMP) echo facility to send ICMP echo request packets to the IP destination that you specify. See Chapter 16 for more information about ICMP.

Ping is implemented for the EME module and the Multilayer Switching Modules of the Switch 4007 system. See Part II of this guide for information about using the Ping feature for the EME module.

When a router sends an echo request packet to an IP station using ping, the router waits for an ICMP echo reply packet. The response indicates whether the remote IP is available, unreachable, or not responding.

Important Consideration

When you specify a hostname with ping, the hostname and its associated IP address must be configured on a network name server. Also, you must add the IP address on the name server to the list of name server addresses that are associated with the network domain name. See Chapter 16**.**

Using Ping

The system provides two ping functions:

- **ping** — Uses the hostname or IP address to ping a host with default options
- **advancedPing** — Uses the hostname or IP address to ping a host with the advanced ping options that you specify

Ping Responses

This list gives the possible responses to a ping:

- If the host is reachable, the system displays information about the ICMP reply packets and the response time to the ping. The amount of information depends on whether the quiet option is enabled or disabled.
- If the host does not respond, the system displays the ICMP packet information and this message: `Host is Not Responding.` (You may see this message if you have not configured your gateway IP address.)
- If the packets cannot reach the host, the system displays the ICMP packet information and this message: `Host is Unreachable.` A host is unreachable when there is no route to that host.

Strategies for Using Ping

Follow these strategies for using ping:

- Ping devices when your network is operating normally so that you have a performance baseline for comparison.
- Ping by *IP address* when:
  - You want to test devices on different subnetworks. This method allows you to ping your network segments in an organized way, rather than having to remember all the hostnames and locations.
  - Your DNS server is down and your system cannot look up host names properly. You can ping with IP addresses even if you cannot access hostname information.
- Ping by *hostname* when you want to identify DNS server problems.
- To troubleshoot problems involving large packet sizes, ping the remote host repeatedly, increasing the packet size each time.

traceRoute

Use the traceRoute feature to track the route of an IP packet through the network. TraceRoute information includes all of the nodes in the network through which a packet passes to get from its origin to its destination. The traceRoute feature uses the IP time-to-live (TTL) field in User Datagram Protocol (UDP) probe packets to elicit an ICMP Time Exceeded message from each gateway to a particular host.

traceRoute is implemented for the Multilayer Switching Modules of the Switch 4007 system.

Using traceRoute

The system provides two traceRoute functions:

- **traceRoute** — Uses the hostname or IP address to trace a route to a host with default options
- **advancedTraceRoute** — Uses the hostname or IP address to trace a route to a host with the advanced traceRoute options that you specify

traceRoute Operation

To track the route of an IP packet, the traceRoute feature launches UDP probe packets with a small TTL value and then listens for an ICMP Time Exceeded reply from a gateway. Probes start with a small TTL of 1 and increase the value by 1 until one of the following events occurs:

- The system receives a Port Unreachable message, indicating that the packet reached the host.
- The probe exceeds the maximum number of hops. The default is `30`.

At each TTL setting, the system launches three UDP probe packets, and the traceRoute display shows a line with the TTL value, the address of the gateway, and the round-trip time of each probe. If a probe answers from different gateways, the traceRoute feature prints the address of each responding system. If no response occurs in the 3-second time-out interval, traceRoute displays an asterisk (*) for that probe. Other characters that can be displayed include the following:

- `!N` — Network is unreachable
- `!H` — Host is unreachable
- `!P` — Protocol is unreachable
- `!F` — Fragmentation is needed
- `!<n>` — Unknown packet type

SNMP

Simple Network Management Protocol (SNMP), one of the most widely used management protocols, allows management communication between network devices and your management workstation across TCP/IP networks.

SNMP is implemented for the EME module, Layer 2 and Multilayer Switching Modules, and the switch fabric module of the Switch 4007 system. See the chapters in Part II of this guide *for information about using the SNMP features implemented for the EME module.*

Most management applications, including Status Watch applications, require SNMP to perform their management functions.

SNMP Overview

The following sections provide an overview of SNMP.

Manager/Agent Operation

SNMP communication requires a *manager* (the station that is managing network devices) and an *agent* (the software in the devices that talks to management station). SNMP provides the language and the rules that the manager and agent use to communicate.

Managers can discover agents:

- Through autodiscovery tools on Network Management Platforms (such as HP OpenView Network Node Manager)
- When you manually enter IP addresses of the devices that you want to manage

For agents to discover their managers, you must provide the agent with the IP address of the management station or stations.

Managers send requests to agents (either to send information or to set a parameter), and agents provide the requested data or set the parameter. Agents can also send information to the managers (without being requested by the managers) through trap messages, which inform the manager that certain events have occurred.

SNMP Messages

SNMP supports queries (called *messages*) that allow the protocol to transmit information between the managers and the agents. Types of SNMP messages:

- **Get** and **Get-next** — The management station requests an agent to report information.
- **Set** — The management station requests an agent to change one of its parameters.
- **Get Responses** — The agent responds to a Get, Get-next, or Set operation.
- **Trap** — The agent sends an unsolicited message informing the management station that an event has occurred.

Management Information Bases (MIBs) define what can be monitored and controlled within a device (that is, what the manager can Get and Set). An agent can implement one or more groups from one or more MIBs. See "Management Information Base (MIB)" later in this chapter for more information.

Trap Reporting

Traps are events that devices generate to indicate status changes. Every agent supports some trap reporting. You must configure trap reporting at the devices so that these events are reported to your management station to be used by the Network Management Platforms (such as HP OpenView Network Node Manager or SunNet Manager).

You do not need to enable all traps to effectively manage a switch. To decrease the burden on the management station and on your network, you can limit the traps reported to the management station.

MIBs are not required to document traps. The SNMP agent supports the limited number of traps defined in Table 96. More traps may be defined in vendors' private MIBs.

Each Layer 2 and Multilayer Switching Module supports a different subset of the traps listed in Table 96. Connect to a module and use the `snmp trap addModify` command to see the list of traps available for that module.

Table 96 Traps Supported by SNMP

Trap No.	Trap Name	Source	Indication
1	Cold Start	MIB II	The agent has started or been restarted.
2	Link Down	MIB II	The status of an attached communication interface has changed from *up* to *down*.
3	Link Up	MIB II	The status of an attached communication interface has changed from *down* to *up*.
4	Authentication Failure	MIB II	The agent received a request from an unauthorized manager.
5	New Root	Bridge MIB	The sending agent has become the new root of the Spanning Tree.
6	Topology Change	Bridge MIB	Any of bridge configured ports transitions from the Learning state to the Forwarding state, or from the Forwarding state to the Blocking state.
7	System Overtemperature	3C System MIB	The system temperature exceeds a certain threshold.
8	Power Supply Failure	3C System MIB	The trap that is generated when a power supply unit fails in a system with a dual power supply.
13	Address Threshold	3C System MIB	The number of addresses stored in the bridge reaches a certain threshold.
14	System Fan Failure	3C System MIB	One of the system fans fails.
15	SMT Hold Condition	3C FDDI MIB	FDDI SMT state either in holding-prm or holding-sec.
16	SMP Peer Wrap Condition	3C FDDI MIB	FDDI SMT connection does not connect to an M-port under DAS mode.
17	MAC Duplicate Address Condition	3C FDDI MIB	A status there are more than one MAC address.
18	MAC Frame Error Condition	3C FDDI MIB	A status the error frames rate reaches a certain threshold.
19	MAC Not Copied Condition	3C FDDI MIB	A status the not copied frames rate reaches a certain threshold.
20	MAC Neighbor Change	3C FDDI MIB	A change in a MAC's upstream neighbor address or downstream neighbor address.
21	MAC Path Change	3C FDDI MIB	A status that the FDDI Path changes.

Table 96 Traps Supported by SNMP (continued)

Trap No.	Trap Name	Source	Indication
22	Port LER Condition	3C FDDI MIB	A status FDDI port link error rate reaches a certain threshold.
23	Port Undesired Connection	3C FDDI MIB	A port connection does not math the connection policy.
24	Port EB Error Condition	3C FDDI MIB	Elasticity Buffer has overflowed.
25	Port Path Change	3C FDDI MIB	Any port path change.
26	Rising Alarm	RMON MIB	An alarm entry crosses its rising threshold.
27	Falling Alarm	RMON MIB	An alarm entry crosses its falling threshold.
28	Response Received	POLL MIB	A disabled device begins responding.
29	Response Not Received	POLL MIB	An enabled device stops responding.
30	Resilient Link Switch Trap	3C Resilient link MIB	This trap is generated in response to either of these conditions: ■ If one of the ports in a resilient link pair changes state, which causes a switchover of the active port. ■ If there was no active port and a port has become active.
31	Resilient Link No Switch Trap	3C Resilient link MIB	This trap is generated when one of the ports in a resilient link pair changes state but does *not* cause a switchover of the active port. If such a switchover occurs, trap 30 is generated.
32	VRRP New Master	VRRP MIB	The sending agent transitioned from *Backup* state to *Master* state.
33	VRRP Authentication Failure	VRRP MIB	A VRRP packet is received from a router whose authentication failed. The authentication failure under this trap is sub-divided under three types: ■ Invalid authentication type ■ Authentication type is valid, but does not match the type configured ■ Authentication type is valid and matches, but has the wrong key

Table 96 Traps Supported by SNMP (continued)

Trap No.	Trap Name	Source	Indication
34	Port Monitor Trap	3C System MIB, Port Monitor Table	This trap is generated when the system has exceeded the excessive collision, multiple collision, late collision, runt packet, or FCS error thresholds. This could be due to a duplex mismatch or a malfunctioning device on the port. Layer 2 modules only.
35	QOS Intruder	QOS MIB	This trap is generated when a user attempts to access a network restricted with a QoS One-Way TCP Filter. The trap contains the following information: ■ Source IP Address ■ Destination IP Address ■ Destination IP Port Number ■ QoS Classifier Number To prevent a denial-of-service (DOS) attack, the system will not generate more than one QOS Intruder trap per second. Thus, an attacker cannot flood a management station with traps or overload a switch's ability to pass or handle messages.

To minimize SNMP traffic on your network, you can implement trap-based polling. Trap-based polling allows the management station to start polling only when it receives certain traps. Your management applications must support trap-based polling for you to take advantage of this feature.

Setting Up SNMP on Your System

Access to system information through SNMP is controlled by community strings. See Part II in this guide for information about configuring the community strings and trap reporting.

You must also assign an IP address to the system Ethernet port, depending on where the management station is attached. See the chapters in Part II of this guide for more information.

You can manage the system using an SNMP-based external management application. This application (called the SNMP manager) sends requests to the system, where they are processed by the internal SNMP agent.

You can gain access to the Remote Monitoring (RMON) capabilities of your system through SNMP applications such as Transcend® Network Control Services software. See "RMON in Your System" later in this chapter for information about the RMON capabilities of your system.

The SNMP agent provides access to the collection of information about your system. (You can view many system-specific settings.) Your views of MIB information differ depending on the system SNMP management method that you choose.

In addition, you can configure a system SNMP agent to send traps to an SNMP manager to report significant events.

Administering SNMP Trap Reporting

For network management applications, you can use the Administration Console to manually administer the trap reporting address information.

- **Displaying Trap Reporting Information** — When you display the trap reporting information, the system displays the various SNMP traps and their currently configured destinations.
- **Configuring Trap Reporting** — You can add new trap reporting destination configurations and modify existing configurations. You can define up to 10 destination addresses and the set of traps that are sent to each destination address.

The trap numbers that you enter direct the system to send the corresponding traps to the destination address when the events occur. No unlisted traps are transmitted.

- **Removing Trap Destinations** — When you remove a destination, no SNMP traps are reported to that destination.

- **Flushing All SNMP Trap Destinations** — When you flush the SNMP trap reporting destinations, you remove all trap destination address information for the SNMP agent.
- **Set SNMP smtProxyTraps** — Controls SNMP's ability to alert you, by means of an SNMP-to-SMT proxy, of a significant event occurring in the FDDI station statistics. (Multilayer Switching Modules only.)

Remote Monitoring (RMON)

This section provides information about Remote Monitoring (RMON) and the RMON-1 and RMON-2 Management Information Base (MIB) groups that are implemented in your system. The following topics are included:

- Overview of RMON
- RMON Benefits
- 3Com Transcend RMON Agents
- RMON in Your System
- RMON-1 Groups
- RMON-2 Groups

RMON is implemented for the Layer 2 and Multilayer Switching Modules and the switch fabric module of the Switch 4007 system.

To manage RMON, you use the IP address that is assigned to the EME. See Chapter 16 for information about managing IP interfaces.

You can gain access to the RMON capabilities of the system through SNMP applications such as Transcend Network Control Services software, not through the serial interface or Telnet. For more information about the details of managing 3Com devices using RMON and Transcend tools, see the user documentation for the 3Com Transcend Network Control Services for Windows suite of applications.

Overview of RMON

RMON provides a way to monitor and analyze a local area network (LAN) from a remote location. The Internet Engineering Task Force (IETF) defines RMON-1 (RMON Version 1) in documents RFC 1271 and RFC 1757; RFC 2021 defines the extension of RMON-1, RMON-2 (RMON Version V2).

A typical RMON implementation has two components:

- **Your system** — Your system's built-in probe functionality examines all the LAN traffic on its segments, and keeps a summary of statistics (including historical data) in its local memory.
- **Management station** — Communicates with your system and collects the summarized data from it. The station can be on a different network from the system and can manage the system's probe function through either in-band or out-of-band connections.

The RMON specification consists almost entirely of the definition of the MIB. The RMON MIB contains standard MIB variables that are defined to collect comprehensive network statistics that alert you to significant network events. If the embedded RMON agent operates full time, it collects data on the correct port when the relevant network event occurs.

RMON Benefits

From a network management console, traditional network management applications poll network devices such as switches, bridges, and routers at regular intervals. The console gathers statistics, identifies trends, and highlights network events. The console polls network devices constantly to determine if the network is within its normal operating conditions.

As network size and traffic levels grow, however, the network management console can become overburdened by the amount of data it must collect. Frequent console polling also generates significant network traffic that itself can create problems for the network.

The RMON implementation in your system offers solutions to both of these problems:

- The system examines the network without affecting the characteristics and performance of the network.
- The system can report by exception rather than by reporting constant or frequent information. That is, the system informs the network management console directly if the network enters an abnormal state. The console can then use more information gathered by the system, such as historical information, to diagnose the abnormal condition.

RMON in Your System

Your system supports RMON as follows:

- **RMON-1 support** — The system software offers full-time embedded RMON support using SNMP for seven RMON-1 groups. (RMON-1 defines 10 groups.)
- **RMON-2 Support** — The system software offers embedded RMON support for seven RMON-2 groups. (RMON-2 defines ten groups.) RMON-2 enables the system RMON feature to see above the MAC layer and monitor traffic based on network-layer protocols and addresses.

The embedded RMON support software cannot receive RMONv2 updates for IP, IPX, and AppleTalk unless you have identified and configured a Virtual LAN (VLAN) protocol type.

3Com Transcend RMON Agents

RMON requires one probe per LAN segment. Because a segment is a portion of the LAN that is separated by a bridge or router, the cost of implementing many probes in a large network can be high.

To solve this problem, 3Com has built an inexpensive RMON probe into the Transcend SmartAgent software in each system. With this probe you deploy RMON widely around the network at a cost of no more than the cost of traditional network monitors.

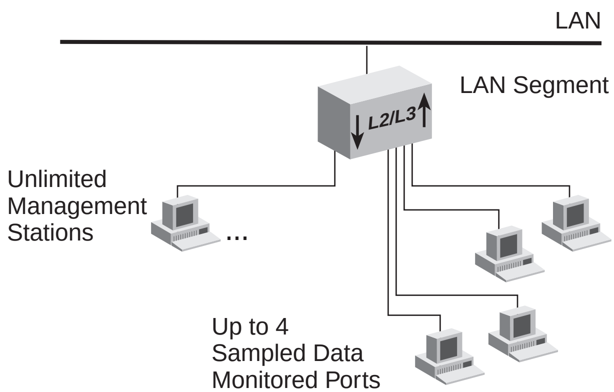
Placing probe functionality inside the system has these advantages:

- You can integrate RMON with normal device management.
- The system can manage conditions proactively.

The system associates statistics with individual ports and then takes action based on these statistics. For example, the system can generate a log event and send an RMON trap if errors on a port exceed a threshold set by the user.

Figure 80 shows an example of the RMON implementation.

Figure 80 Embedded RMON Implemented on the System



Important Considerations

- To manage RMON, you must assign an IP address to the system through the EME management module. See the *Switch 4007 Enterprise Management Engine User Guide* for information about how to do this.
- The system will always keep RMON statistics (group 1) data on all ports.
- The system will keep RMON history (group 2), alarm (group 3), and event (group 9) data on as many ports as its resources allow.
- Multilayer Switching Modules support additional RMON-1 and RMON-2 groups.
  - The system will keep as much protocolDir (group 11), protocolDist (group 12), and probeConfig (group 19) data as its resources will allow.
  - All other RMON group data is hardware sampled. The system can be configured to keep hardware-sampled RMON group data on up to four ports per module.
- RMON data for Gigabit Ethernet is supported on Layer 2 modules.
- RMON data is *not* supported on Multilayer Gigabit Ethernet modules except for the 4-port Gigabit Ethernet Switching module (3CB9RG4). This module supports the statistics, history, alarm, and event groups.
- There is no limit to the number of network management stations monitoring the data.

RMON-1 Groups

The system supports seven of the RMON-1 groups that the IETF defines. Table 97 briefly describes these groups.

The Layer 2 Switching Modules and switch fabric modules support four RMON-1 groups: groups 1, 2, 3, and 9. Multilayer Switching Modules support seven groups: 1, 2, 3, 4, 5, 6, and 9.

Table 97 RMON-1 Groups Supported in the System

RMON-1 Group	Group Number	Purpose
Statistics	1	Maintains utilization and error statistics for the segment being monitored
History	2	Gathers and stores periodic statistical samples from the statistics group
Alarm	3	Allows you to define thresholds for any MIB variable and trigger alarms
Host	4	Discovers new hosts on the network by keeping a list of source and destination physical addresses that are seen in good packets
HostTopN	5	Allows you to prepare reports that describe the hosts that top a list sorted by one of their statistics
Matrix	6	Stores statistics for conversations between pairs of addresses
Event	9	Allows you to define actions (generate traps, log alarms, or both) based on alarms

The system also supports the RMON/FDDI extension groups that the AXON Enterprise-specific MIB specifies. See Table 98.

Table 98 RMON/FTable 98DDI Extension Groups

Group	Group Number	Purpose
axFddiStatistics	axFddi group 1	Maintains utilization and error statistics for the monitored segment
axFddiHistory	axFddi group 2	Gathers and stores periodic statistical samples from the statistics group

Statistics and axFddiStatistics Groups

The statistics and axFDDIStatistics groups record frame statistics for Ethernet and FDDI interfaces. The information available per interface segment includes:

- Number of received octets
- Number of received packets
- Number of received broadcast packets
- Number of received multicast packets
- Number of received packets with CRC or alignment errors
- Number of received packets that are undersized but otherwise well-formed
- Number of received packets that are oversized but otherwise well-formed
- Number of received undersized packets with either a CRC or an alignment error
- Number of detected transmit collisions

Byte sizes include the 4-byte FCS, but exclude the framing bits. Table 99 lists the ethernet packet length counters that are implemented in the RMON-1 statistics group to keep track of the frame sizes that are encountered.

Table 99 Supported Frame Sizes for Ethernet and FDDI

Frame Lengths (Bytes)	
Ethernet	**FDDI**
64	22 or fewer
65 - 127	23 - 63, 64 - 127
128 - 511	128 - 511
512 - 1023	512 - 1023
1024 - 1518 (1024 - 1522 bytes when tagging is enabled)	1024 - 2047, 2048 - 4095

History and axFDDIHistory Groups

The history and axFDDIHistory groups record periodic statistical samples for Ethernet and FDDI interfaces and store them for later retrieval. The information available per interface for each time interval includes:

- Number of received octets
- Number of received packets
- Number of received broadcast packets
- Number of received multicast packets
- Number of received packets with CRC or alignment errors
- Number of received packets that are undersized but otherwise well-formed
- Number of received packets that are oversized but otherwise well-formed
- Number of received undersized packets with either a CRC or an alignment error
- Number of detected transmit collisions
- Estimate of the mean physical layer network utilization

Alarm Group

The system supports the following RMON alarm mechanisms:

- Counters
- Gauges
- Integers
- Timeticks

These RMON MIB objects yield alarms when the network exceeds predefined limits. The most frequently used objects are *counters*, although the other objects may be used in much the same way. The balance of this chapter illustrates RMON functions using counters.

Counters hold and update the number of times an event occurs on a port, module, or switch. *Alarms* monitor the counters and report when counters exceed their set threshold.

Counters are useful when you compare their values at specific time intervals to determine rates of change. The time intervals can be short or long, depending on what you measure.

Occasionally, counters can produce misleading results. Because counters are finite, they are useful for comparing rates. When counters reach a predetermined limit, they *roll over* (that is, return to 0). A single low counter value may accurately represent a condition on the network. On the other hand, the same value may simply indicate a rollover.

When you disable a port, the application may not update some of its associated statistics counters.

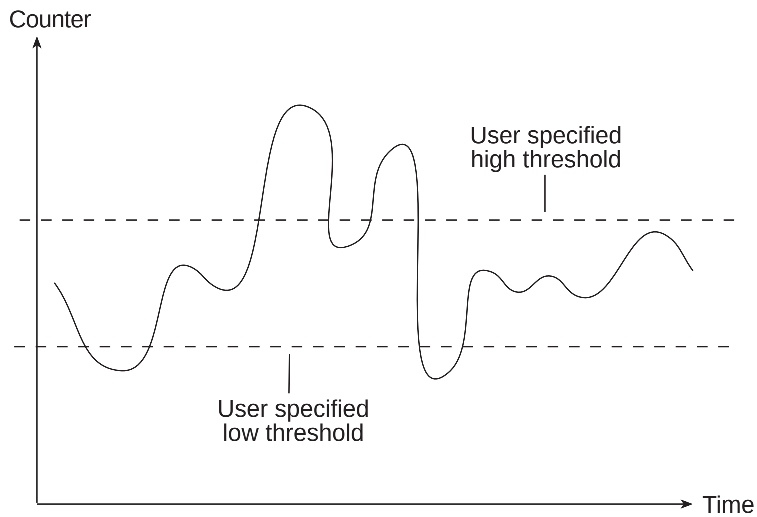
An alarm calculates the difference in counter values over a set time interval and remembers the high and low values. When the value of a counter exceeds a preset threshold, the alarm reports this occurrence.

Using Transcend Network Control Services or any other SNMP network management application, you can assign alarms to monitor any counter, gauge, timetick, or integer. See the documentation for your management application for details about setting up alarms.

Setting Alarm Thresholds Thresholds determine when an alarm reports that a counter has exceeded a certain value. You can set alarm thresholds manually through the network, choosing any value for them that is appropriate for your application. The network management software monitors the counters and thresholds continually during normal operations to provide data for later calibration.

Figure 81 shows a counter with thresholds set manually.

Figure 81 Manually Set Thresholds



You can associate an alarm with the high threshold, the low threshold, or both. The actions that occur because of an alarm depend on the network management application.

RMON Hysteresis Mechanism The RMON hysteresis mechanism prevents small fluctuations in counter values from causing alarms. Alarms occur only when either:

- The counter value exceeds the high threshold after previously falling below the low threshold. (An alarm does not occur if the value has not fallen below the low threshold before rising above the high threshold.)
- The counter value falls below the low threshold after previously exceeding the high threshold. (An alarm does not occur if the value has not first risen above the high threshold.)

For example, in Figure 81, an alarm occurs the first time that the counter exceeds the high threshold, but not the second time. At the first instance, the counter is rising from below the low threshold. In the second instance, the counter is not rising from below the low threshold.

Host Group

The host group records the following statistics for each host (the host group detects hosts on the network by their physical MAC addresses):

- Number of received packets
- Number of transmitted packets
- Number of received octets
- Number of transmitted octets
- Number of transmitted broadcast packets
- Number of transmitted multicast packets

These statistics, indexed by relative order in which the hosts are discovered, appear in *hostTimeTable*.

HostTopN Group

The HostTopN group reports on hosts that top a list that is sorted in order of one of their statistics. Information from this group includes:

- Number of received packets
- Number of transmitted packets
- Number of received octets
- Number of transmitted octets
- Number of transmitted broadcast packets
- Number of transmitted multicast packets

Matrix Group

The matrix group records the following statistics about conversations between sets of addresses:

- Number of packets transmitted from the source address to the destination address
- Number of octets, excluding errors, transmitted from the source address to the destination address
- Number of bad packets transmitted from the source address to the destination address

Event Group

The event group logs alarms or traps network event descriptions. Although alarm group thresholds trigger most events, other RMON groups may define event conditions.

RMON-2 Groups

The system software supports seven RMON-2 groups defined by the IETF in RFC 2021 and one object from the probe configuration group. Table 100 briefly describes these groups.

The RMON-2 groups are supported on Multilayer Switching Modules only.

Table 100 RMON-2 Groups Supported in the System

RMON-2 Group	Group Number	Purpose
protocolDir	11	Provides a list of all protocols that the probe can interpret (protocols for which the probe can decode and count packets). The protocols can be different network layer, transport layer, and higher layer protocols. This group allows the addition, deletion, and configuration of entries in the list.
protocolDist	12	Maintains a table of aggregate statistics on the amount of traffic that each protocol generates per LAN segment (not for each host or application running on each host).
AddressMap	13	Maintains a table that maps each network address to a specific MAC address and port on an attached device and the physical address on the subnetwork
nlHost	14	Provides network-layer host statistics on the amount of traffic going in and out of hosts based on network-layer address
nlMatrix	15	A network-layer matrix that provides statistics on the amount of traffic between source/destination pairs of hosts based on network-layer address. It also maintains a TopN table to rank pairs of hosts based on the number of octets or number of packets sent between pairs of hosts.
alHost	16	Traffic statistics to and from each host by application layer. Same as nlHost except that traffic broken down by protocols can be recognized by ProtocolDir

Table 100 RMON-2 Groups Supported in the System (continued)

RMON-2 Group	Group Number	Purpose
alMatrix	17	Traffic statistics on conversations between pairs of hosts, segmented by application layer protocol.
probeConfig. probeCapabilities	19	Defines standard parameters that control the configuration of RMON probe functionality. The system currently supports only the probeCapabilities object from this group.

Protocol Directory Group

The protocolDir group provides information about the protocols that a particular RMON probe has or can interpret. It provides a common method of storing information about the protocols and makes it easier for a manager to monitor traffic above the MAC layer.

This group features a protocol directory table that has an entry for each protocol. This enables the probe to decode and count protocol data units (PDUs). Information in the table includes the following:

- A protocol identifier (a unique octet string for a specific protocol)
- Protocol parameters (information about the probe's capabilities for a specific protocol)

Protocol Distribution Group

The Switch 4007 provides protocolDist segmentation only between the IP, IPX, and Appletalk protocols.

The protocolDist group tracks how many octets and packets the supported protocols have sent. It features two tables, a protocol distribution control table that manages the collection of the statistics for the supported protocols, and a protocol distribution statistics table that records the statistics. In the control table, each row represents a network interface associated with the probe and controls rows in the statistics table (a row for each protocol associated with the interface).

The protocol distribution statistics table includes the following statistics:

- The number of packets received for each protocol
- The number of octets transmitted to this address since it was added to the network-layer host table.

Address Map Group

The addressMap group maps each network address to a specific MAC-level address and to a specific port on the network device. This group provides three scalar objects (to track address-mapping entry insertions, deletions, and the maximum number of entries), an address map control table, and an address map data table.

Unlike other RMON control tables and data tables, the address map data table is not indexed by a row of the control table. The data table has entries that enable the mapping between the network addresses (normally IP addresses) and MAC addresses. The control table has an entry for each subnetwork connected to the probe so that addresses can be discovered on a new subnetwork and address mapping entries can be placed in the data table.

Network-Layer Host Group

The nlHost group gathers statistics about packets based on their network-layer address. (The RMON-1 host group gathers statistics based on MAC address.)

This group features a host control table and a host data table.

Network-Layer Matrix Group

The nlMatrix group gathers statistics about pairs of hosts based on network-layer address. (The RMON-1 matrix group gathers statistics based on MAC address.)

This group features two control tables and three data tables. One control table and its data tables collect matrix statistics; the other control table and its data table collect TopN statistics (reports describing hosts that top a list).

Application-Layer Host Group

The alHost group gathers statistics about IP packets over a monitored port based on their protocol. (The RMON-2 network-layer matrix group gathers statistics based on the network address).

This group features a host data table.

Application-Layer Matrix Group

The alMatrix group gathers statistics about pairs of hosts conversing over a monitored port based on protocol. (The RMON-2 network-layer matrix group gathers statistics based on the network address).

This group features one control table and three data tables. The alMatrix SD and alMatrix DS tables monitor traffic flows per conversation over monitored ports. The control table and its data table collect TopN statistics (reports describing hosts that top a list).

Probe Configuration Group Capabilities

The probeConfig group outlines a standard set of configuration parameters for RMON probes. Currently, your system supports one object in the probeConfig group, the probeCapabilities object. The function of this object is to identify the RMON groups that the probe supports.

Management Information Base (MIB)

This section provides information on the Management Information Base (MIB). A MIB is a structured set of data that describes the way that the network is functioning. The management software, known as the *agent*, gains access to this set of data and extracts the information it needs. The agent can also store data in the MIB. The following topics are covered:

- MIB Files
- Compiler Support
- MIB Objects
- MIB Tree
- MIB-II
- RMON-1 MIB
- RMON-2 MIB
- 3Com Enterprise MIBs

MIB II is implemented for the EME module, Layer 2 and Multilayer Switching Modules, and the switch fabric module of the Switch 4007 system. See the Part II in this guide *for information about using the MIB II features that are implemented for the EME module.*

MIB Files

The organization of a MIB allows a Simple Network Management Protocol (SNMP) network management package, such as the Transcend Network Control Services application suite, to manage a network device without having a specific description of that device. 3Com ships the following MIB files with Extended System software as ASN.1 files.

- **BRIDGE-MIB.mib** — Bridge MIB, RFC 1493. Layer 2 and Layer 3.

  Unsupported groups and tables in this MIB:

  - dot1dSr group (for Layer 3, see SOURCE-ROUTING-MIB.mib below)
  - dot1dStatic group
- **ENTITY-MIB.mib** — RFC 2037. Layer 2, Layer 3, and EME.
- **ETHERNET.mib** — Ethernet MIB, RFC 1398. Layer 2 and Layer 3.
- **FDDI-SMT73-MIB.mib** — FDDI SMT 7.3 MIB, RFC 1512. Layer 3 only.
  - dot3CollTable
  - dot3Test group
  - dot3Errors group
  - dot3ChipSets group
- **FDDI-MIB.mib** — FDDI Station Management MIB, RFC 1285. Layer 3 only.
- **IANAifType-MIB-V1SMI.mib** — Internet Assigned Numbers Authority MIB, SMI Version 1, RFC 1573. Layer 2 and Layer 3.
- **IF-MIB-V1SMI.mib** — Interface MIB, Version 1, RFC 1573. Layer 2 and Layer 3.

  Unsupported tables in this MIB:

  - ifTestTable
  - ifRcvAddressTable
  - ifHC 64-bit counters
- **MIB2-MIB.mib** — MIB-II MIB, RFC 1213. Layer 2, Layer 3, and EME.

  Unsupported groups and tables in this MIB:

  - egp group
- **OSPF-MIB.mib** — Open Shortest Path First MIB, RFC 1850. Layer 3 only.
- **RMON-MIB.mib** — RMON MIB, RFC 1757. Layer 2 and Layer 3.

RMON statistics for Gigabit Ethernet are supported on Layer 2 modules. The 4-Port Gigabit Ethernet Layer 3 Switching Module (GBIC) supports the Statistics, History, Alarm, and Event groups.

Supported groups in this MIB:

- statistics (Layers 2 and 3)
- history (Layers 2 and 3)
- alarm (Layers 2 and 3)
- hosts (Layer 3)
- hostTopN (Layer 3)
- matrix (Layer 3)
- event (Layers 2 and 3)

- **axonFddiRmon.mib** — AXON RMON MIB, proprietary support. Layer 3 only. On FDDI modules, these replace the RMON-1 statistics and history groups. FDDI modules support all other RMON-1 and RMON-2 groups.
  - axFddiStatistics
  - axFddiHistory
- **RMON2-MIB-V1SMI.mib** — RMON v2, SMI Version 1 MIB, RFC 2021. Layer 3 only.
  - protocolDir (RMONv2)
  - protocolDist (RMONv2)
  - addressMap (RMONv2)
  - nlHost (RMONv2)
  - nlMatrix (RMONv2)
  - alHost (RMONv2)
  - alMatrix (RMONv2)
  - probeCapabilities object of probeConfig group (RMONv2)

A maximum of four different ports can be configured for the following RMON groups at any given time:

- addressMap
- alHost
- alMatrix
- hosts
- hostTopN
- matrix
- nlHost
- nlMatrix

- **SNMPv2-MIB.mib** — Used by other MIBs, RFC 1907. Layer 2 and Layer 3.
- **SOURCE-ROUTING-MIB.mib** — Source Routing MIB, RFC 1525. Layer 3 only.
- **VRRP-MIB.mib** — Virtual Router Redundancy Protocol MIB, Draft RFC. Layer 3 only.
- **3Com Enterprise MIBs** — See “3Com Enterprise MIBs” later in this chapter.

Compiler Support

Compiler Support ASN.1 MIB files are provided for these MIB compilers:

- SunNet Manager (version 2.0)
- SMICng (version 2.2.06)

MIB Objects

The data in the MIB consists of objects that represent features of the equipment that an agent can control and manage. Examples of objects in the MIB include a port that you can enable or disable and a counter that you can read.

A counter is a common type of MIB object used by RMON. A counter object may record the number of frames transmitted onto the network. The MIB may contain an entry for the counter object something like the one in Figure 82.

Figure 82 Example of an RMON MIB Counter Object

```
etherStatsPkts  OBJECT-TYPE
          SYNTAX              Counter
          ACCESS              read-only
          STATUS              mandatory
          DESCRIPTION
                    This is a total number of packets
                    received, including bad packets,
                    broadcast packets, and multicast
                    packets.
          ::= {  etherStatsEntry 5 }
```

The counter object information includes these items:

- The name of the counter. In Figure 82, the counter is called *etherStatsPkts* (Ethernet, Statistics, Packets).
- Access level. In Figure 82, access is read-only.
- The number of the counter's column in the table. In Figure 82, the counter is in column 5 of the *etherStatsEntry* table.

The name of the table where the counter resides is *3CometherStatTable,* although this name does not appear in the display.

To manage a network, you do not need to know the contents of every MIB object. Most network management applications, including Transcend Network Control Services, make the MIB transparent. However, by knowing how different management features are derived from the MIB you can better understand how to use the information they provide.

MIBs include MIB-II, other standard MIBs (such as the RMON MIB), and vendors' private MIBs (such as enterprise MIBs from 3Com). These MIBs and their objects are part of the MIB tree.

MIB Tree

The MIB tree is a structure that groups MIB objects in a hierarchy and uses an abstract syntax notation (ASN.1) to define manageable objects. Each item on the tree is assigned a number (shown in parentheses after each item), which creates the path to objects in the MIB. See Figure 83. This path of numbers is called the object identifier (OID). Each object is uniquely and unambiguously identified by the path of numeric values.

When the system software performs an SNMP Get operation, the management application sends the OID to the agent, which in turn determines if the OID is supported. If the OID is supported, the agent returns information about the object.

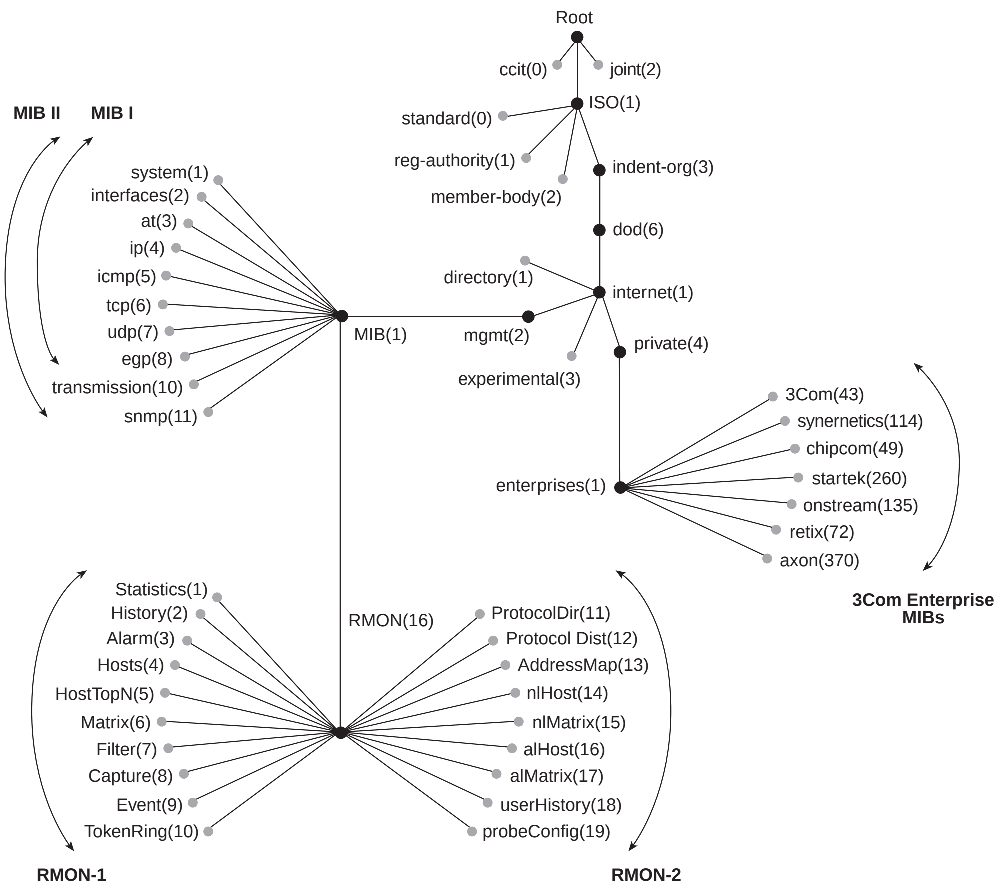
For example, to retrieve an object from the RMON MIB, the software uses this OID:

```
1.3.6.1.2.1.16
```

which indicates this path:

```
iso(1).indent-org(3).dod(6).internet(1).mgmt(2).mib(1)
.RMOM 16)
```

Figure 83 MIB Tree Showing Key MIBs



MIB-II

MIB-II defines various groups of manageable objects that contain device statistics as well as information about the device, device status, and the number and status of interfaces.

The MIB-II data is collected from network devices using SNMP. As collected, this data is in its raw form. To be useful, data must be interpreted by a management application, such as Status Watch.

MIB-II, the only MIB that has reached Internet Engineering Task Force (IETF) standard status, is the one MIB that all SNMP agents are likely to support.

Table 101 lists the MIB-II object groups. The number following each group indicates the group's branch in the MIB subtree.

MIB-I supports groups 1 through 8; MIB-II supports groups 1 through 8, plus two additional groups.

Table 101 MIB-II Group Descriptions

MIB-II Group	Purpose
system(1)	Operates on the managed node
interfaces(2)	Operates on the network interface (for example, a port or MAC) that attaches the device to the network
at(3)	Were used for address translation in MIB-I but are no longer needed in MIB-II
ip(4)	Operates on the Internet Protocol (IP)
icmp(5)	Operates on the Internet Control Message Protocol (ICMP)
tcp(6)	Operates on the Transmission Control Protocol (TCP)
udp(7)	Operates on the User Datagram Protocol (UDP)
egp(8)	Operates on the Exterior Gateway Protocol (EGP)
transmission(10)	Applies to media-specific information (implemented in MIB-II only)
snmp(11)	Operates on SNMP (implemented in MIB-II only)

RMON-1 MIB

RMON-1 is a MIB that enables the collection of data about the network itself, rather than about devices on the network.

The IETF definition for the RMON-1 MIB specifies several groups of information. These groups are described in Table 102.

Table 102 RMON-1 Group Descriptions

RMON-1 Group	Description
Statistics(1)	Total LAN statistics
History(2)	Time-based statistics for trend analysis
Alarm(3)	Notices that are triggered when statistics reach predefined thresholds
Hosts(4)	Statistics stored for each station's MAC address
HostTopN(5)	Stations ranked by traffic or errors
Matrix(6)	Map of traffic communication among devices (that is, who is talking to whom)
Filter(7)	Packet selection mechanism
Capture(8)	Traces of packets according to predefined filters
Event(9)	Reporting mechanisms for alarms
Token Ring(10)	■ Ring Station — Statistics and status information associated with each token ring station on the local ring, which also includes status information for each ring being monitored ■ Ring Station Order — Location of stations on monitored rings ■ Source Routing Statistics — Utilization statistics derived from source routing information optionally present in token ring packets

RMON-2 MIB

RMON-1 and RMON-2 are complementary MIBs. The RMON-2 MIB extends the capability of the original RMON-1 MIB to include protocols above the MAC level. Because network-layer protocols (such as IP) are included, a probe can monitor traffic through routers attached to the local subnetwork.

Use RMON-2 data to identify traffic patterns and slow applications. The RMON-2 probe can monitor:

- The sources of traffic arriving by a router from another network
- The destination of traffic leaving by a router to another network

Because it includes higher-layer protocols (such as those at the application level), an RMON-2 probe can provide a detailed breakdown of traffic by application.

Table 103 shows the additional MIB groups available with RMON-2.

Table 103 RMON-2 Group Descriptions

RMON-2 Group	Description
Protocol Directory(11)	Lists the inventory of protocols that the probe can monitor
Protocol Distribution(12)	Collects the number of octets and packets for protocols detected on a network segment
Address Map(13)	Lists MAC-address-to-network-address bindings discovered by the probe, and the interface on which the bindings were last seen
Network Layer Host(14)	Counts the amount of traffic sent from and to each network address discovered by the probe
Network Layer Matrix(15)	Counts the amount of traffic sent between each pair of network addresses discovered by the probe
Application Layer Host(16)	Counts the amount of traffic, by protocol, sent from and to each network address discovered by the probe
Application Layer Matrix(17)	Counts the amount of traffic, by protocol, sent between each pair of network addresses discovered by the probe
User History(18)	Periodically samples user-specified variables and logs the data based on user-defined parameters
Probe Configuration(19)	Defines standard configuration parameters for RMON probes

3Com Enterprise MIBs

3Com Enterprise MIBs allow you to manage unique and advanced functionality of 3Com devices. These MIBs are shipped with your system. Figure 83 shows some of the 3Com Enterprise MIB names and numbers. The following MIBs are included in 3Com(43).

- **cb9000.mib** — Top-level 9000 MIB. EME.
- **cb9000Mod.mib** — Module-specific information. Layer 2 and Layer 3.
- **cb9eme.mib** — EME-specific objects. EME.
- **3cFddi.mib** — 3Com FDDI MIB (43.29.10). Layer 3 only.
- **3cFilter.mib** — 3Com Packet Filtering MIB, standard and custom (43.29.4.20). Layer 3 only.
- **3cigmpSnoop.mib** — 3Com IGMP Snooping MIB. Layer 2 only.
- **3com0304.mib** — 3Com Resilient Links MIB (43.10.15). Layer 2 and Layer 3.
- **3cPolicy.mib** — 3Com Policy Management MIB (43.29.4.23). Layer 3 only.
- **3cPoll.mib** — 3Com Remote Polling MIB (43.29.4.22). Layer 2 and Layer 3.
- **3cProd.mib** — 3Com Transcend Product Management MIB (43.1). Layer 2, Layer 3, and EME.
- **3cQos.mib** — 3Com QoS MIB (43.29.4.21). Layer 3 only.
- **3cSys.mib** — 3Com System MIB (43.29.4). Layer 2 and Layer 3.

  Unsupported groups in this MIB:

  - a3ComSysSlot
  - a3ComSysControlPanel
  - a3ComSysSnmp
- **3cSysBridge.mib** — 3Com Bridging MIB (43.29.4.10). Layer 2 and Layer 3.
- **3cSysFt.mib** — 3Com File Transfer MIB (43.29.4.14). Layer 2 and Layer 3.
- **3cSysSmt.mib** — 3Com SMT MIB (43.29.4.9). Layer 3 only.
- **3cTrunk.mib** — 3Com Port Trunking MIB (43.10.1.15.1). Layer 2 and Layer 3.

- **3cVlan.mib** — 3Com VLAN MIB (43.10.1.14.1). Layer 2 and Layer 3.
- **3cWeb.mib** — 3Com Web Management MIB (43.29.4.24). Layer 2 and Layer 3.

MIB names and numbers are usually retained when organizations restructure their businesses; therefore, some of the 3Com Enterprise MIB names may not contain the word "3Com."

IV REFERENCE

A TECHNICAL SUPPORT

3Com provides easy access to technical support information through a variety of services. This appendix describes these services.

Information contained in this appendix is correct at time of publication. For the most recent information, 3Com recommends that you access the 3Com Corporation World Wide Web site.

Online Technical Services

3Com offers worldwide product support 24 hours a day, 7 days a week, through the following online systems:

- World Wide Web Site
- 3Com FTP Site
- 3Com Bulletin Board Service
- 3Com Facts Automated Fax Service

World Wide Web Site

To access the latest networking information on the 3Com Corporation World Wide Web site, enter this URL into your Internet browser:

`http://www.3com.com/`

This service provides access to online support information such as technical documentation and software, as well as support options that range from technical education to maintenance and professional services.

3Com FTP Site

Download drivers, patches, software, and MIBs across the Internet from the 3Com public FTP site. This service is available 24 hours a day, 7 days a week.

To connect to the 3Com FTP site, enter the following information into your FTP client:

- Hostname: **`ftp.3com.com`** (or **`192.156.136.12`**)
- Username: **`anonymous`**
- Password: **`<your Internet e-mail address>`**

You do not need a user name and password with Web browser software such as Netscape Navigator and Internet Explorer.

3Com Bulletin Board Service

The 3Com BBS contains patches, software, and drivers for 3Com products. This service is available through analog modem or digital modem (ISDN) 24 hours a day, 7 days a week.

Access by Analog Modem

To reach the service by modem, set your modem to 8 data bits, no parity, and 1 stop bit. Call the telephone number nearest you:

Country	Data Rate	Telephone Number
Australia	Up to 14,400 bps	61 2 9955 2073
Brazil	Up to 14,400 bps	55 11 5181 9666
France	Up to 14,400 bps	33 1 6986 6954
Germany	Up to 28,800 bps	4989 62732 188
Hong Kong	Up to 14,400 bps	852 2537 5601
Italy	Up to 14,400 bps	39 2 27300680
Japan	Up to 14,400 bps	81 3 3345 7266
Mexico	Up to 28,800 bps	52 5 520 7835
P.R. of China	Up to 14,400 bps	86 10 684 92351
Taiwan, R.O.C.	Up to 14,400 bps	886 2 377 5840
U.K.	Up to 28,800 bps	44 1442 438278
U.S.A.	Up to 53,333 bps	1 847 262 6000

Access by Digital Modem

ISDN users can dial in to the 3Com BBS using a digital modem for fast access up to 64 Kbps. To access the 3Com BBS using ISDN, call the following number:

1 847 262 6000

3Com Facts Automated Fax Service

The 3Com Facts automated fax service provides technical articles, diagrams, and troubleshooting instructions on 3Com products 24 hours a day, 7 days a week.

Call 3Com Facts using your Touch-Tone telephone:

1 408 727 7021

Support from Your Network Supplier

If you require additional assistance, contact your network supplier. Many suppliers are authorized 3Com service partners who are qualified to provide a variety of services, including network planning, installation, hardware maintenance, application training, and support services.

When you contact your network supplier for assistance, have the following information ready:

- Product model name, part number, and serial number
- A list of system hardware and software, including revision levels
- Diagnostic error messages
- Details about recent configuration changes, if applicable

If you are unable to contact your network supplier, see the following section on how to contact 3Com.

Support from 3Com

If you are unable to obtain assistance from the 3Com online technical resources or from your network supplier, 3Com offers technical telephone support services. To find out more about your support options, call the 3Com technical telephone support phone number at the location nearest you.

When you contact 3Com for assistance, have the following information ready:

- Product model name, part number, and serial number
- A list of system hardware and software, including revision levels
- Diagnostic error messages
- Details about recent configuration changes, if applicable

Here is a list of worldwide technical telephone support numbers:

Country	Telephone Number	Country	Telephone Number
Asia Pacific Rim			
Australia	1 800 678 515	P.R. of China	10800 61 00137 or
Hong Kong	800 933 486		021 6350 1590
India	61 2 9937 5085	Singapore	800 6161 463
Indonesia	001 800 61 009	S. Korea	
Japan	0031 61 6439	From anywhere in S. Korea:	82 2 3455 6455
Malaysia	1800 801 777	From Seoul:	00798 611 2230
New Zealand	0800 446 398	Taiwan, R.O.C.	0080 611 261
Pakistan	61 2 9937 5085	Thailand	001 800 611 2000
Philippines	1235 61 266 2602		
Europe			
From anywhere in Europe, call:	+31 (0)30 6029900 phone		
	+31 (0)30 6029999 fax		
From the following European countries, you may use the toll-free numbers:			
Austria	06 607468	Netherlands	0800 0227788
Belgium	0800 71429	Norway	800 11376
Denmark	800 17309	Poland	0800 3111206
Finland	0800 113153	Portugal	05 05313416
France	0800 917959	South Africa	0800 995014
Germany	0130 821502	Spain	900 983125
Hungary	00800 12813	Sweden	020 795482
Ireland	1 800 553117	Switzerland	0800 55 3072
Israel	177 3103794	U.K.	0800 966197
Italy	1678 79489		
Latin America			
Argentina	AT&T +800 666 5065	Mexico	01 800 CARE (01 800 2273)
Brazil	0800 13 3266	Peru	AT&T +800 666 5065
Chile	1230 020 0645	Puerto Rico	800 666 5065
Colombia	98012 2127	Venezuela	AT&T +800 666 5065
North America	1 800 NET 3Com (1 800 638 3266)		
	Enterprise Customers 1 800 876-3266		

Returning Products for Repair

Before you send a product directly to 3Com for repair, you must first obtain a Return Materials Authorization (RMA) number. Products sent to 3Com without RMA numbers will be returned to the sender unopened, at the sender’s expense.

To obtain an RMA number, call or fax:

Country	Telephone Number	Fax Number
Asia, Pacific Rim	65 543 6500	65 543 6348
Europe, South Africa, and Middle East	+ 44 1442 435860	+ 44 1442 435718
From the following European countries, you may call the toll-free numbers; select option 2 and then option 2:		
Austria	06 607468	
Belgium	0800 71429	
Denmark	800 17309	
Finland	0800 113153	
France	0800 917959	
Germany	0130 821502	
Hungary	00800 12813	
Ireland	1800553117	
Israel	177 3103794	
Italy	1678 79489	
Netherlands	0800 0227788	
Norway	800 11376	
Poland	00800 3111206	
Portugal	05 05313416	
South Africa	0800 995014	
Spain	900 983125	
Sweden	020 795482	
Switzerland	0800 55 3072	
U.K.	0800 966197	
Latin America	1 408 326 2927	1 408 326 3355
U.S.A. and Canada	1 800 NET 3Com (1 800 638 3266)	1 408 326 7120
	Enterprise Customers 1 800 876-3266	

INDEX

B

C

H

I

P